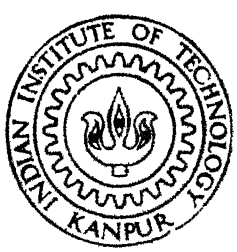
DIMENSIONALITY REDUCTION TECHNIQUES FOR INTELLIGENT SYSTEMS

by
Manish Kumar Shrivastava



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
MARCH, 1998

DIMENSIONALITY REDUCTION TECHNIQUES FOR INTELLIGENT SYSTEMS

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
Master of Technology

by

Manish Kumar Shrivastava

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
March 1998

CERTIFICATE

This is to certify that the work contained in thesis entitled " **Dimensionality Reduction Techniques for Intelligent Systems** " by Mr. Manish Kumar Shrivastava has been carried out under my supervision and that has not been submitted elsewhere for a degree.

Dr. Prem K. Kalra
Professor
Department of Electrical Engineering
Indian Institute of Technology , Kanpur

ACKNOWLEDGMENT

It is my great pleasure to express my deepest gratitude to Dr.P.K.Kalra for his inspiring guidance and patient supervision throughout the programme. Besides the academic guidance, he always provided me moral support also, when things appeared difficult to me. I am also grateful to him for the tremendous lab facilities and free work culture he provided.

I wish to thank all my friends and well-wishers who made my stay at I.I.T. Kanpur all these years a memorable and pleasant experience. The unstinting help rendered to me by my friends Neelamadhab Biswal, V. Sharat Chandra, Basudeb Munshi and B.K. Bhujabal.

I have no words to express my thanks to my parents who have always been a constant source of inspiration to me. To say the least, their magnanimity, benevolence, moral support and encouragement has been to a very great extent instrumental in my work.

Dedicated To
my parents

ABSTRACT

The demand for the large number of instances grows exponentially with dimensionality of feature space. In addition, greater computational complexity is implied by higher dimensionality. Various modelling techniques face such problems. So before inputting the samples or instances for modelling, it is always preferred to preprocess the available data set, which include data prioritisation and feature clustering.

Principal Component Analysis (PCA) and Independent Component Analysis (ICA) are two commonly used data prioritisation techniques, which absorb most of data variation with smaller dimensions. In PCA, we have uncorrelated components and in ICA, we have independent components.

Heterogeneity is always present in the data, which also offers problems in modelling. Clustering or grouping of samples (generally features based) is an answer to such problems. K-means clustering, partitions the total data set in to k classes. Sometimes, in K-means clustering, there may be some classes which are empty. Such problem is taken care in fuzzy C-means clustering, in which each sample belongs to all c classes with some membership. A class of neural networks also performs clustering and classification. Kohenon self organising map performs similar action to K-means method. But, two unsupervised networks ART2 and Fuzzy ART classify the samples depending up on a factor called vigilance factor. The advantage of these networks is that, we need not to specify the number of classes in advance. Inductive reasoning (ID3) can also be applied for this purpose.

All these techniques have been applied to steel converter lining life prediction problem. Using PCA and ICA, we could reduce the dimension of the system to 15×13 from 15×26. Then these 15×13 systems are given as input to various modelling techniques. Generally, ICA is giving good results. In clusterwise modelling, K-means, Fuzzy C-means and kohenon clustering give better results.

Contents

List Of Tables

CHAPTER 1

INTRODUCTION

The demand for large number of instances grows exponentially with the dimensionality of the feature space. In addition, greater computational complexity is implied by higher dimensionality. A variety of methods for dimensionality reduction have been proposed, such as following:

- *Principal Component Analysis*
- *Independent Component Analysis*
- *Feature Clustering*

The first two approaches reduce dimensionality by either prioritising the features or by forming the linear combinations of them. The third approach merges feature which are highly correlated since they provide redundant information.

This section presents a brief idea about Principal Component Analysis (PCA), Independent Component Analysis (ICA) and various clustering algorithms such as K-means clustering, Fuzzy C-means clustering, Kohenon Self Organizing Map, ART2, Fuzzy ART and classification tree method such as ID3.

1.1 Brief review of techniques used

(i) Principal Component Analysis

Principal Component Analysis is concerned with the variance - covariance structure of the multivariate system. This is most widely used method for data reduction. Actually, PCA is related with the eigenvectors of the covariance (correlation) matrix of the given data set. The associated eigenvalue represents the percentage loading of this particular component on the system. Each PC is related with some features, which are dominant in the direction described by it. A threshold level for cumulative loading of components can be fixed according to the application. As we achieve this threshold we can stop considering further components. Then we select the most dominant feature for each successive PC. These many features are retained and rest are deleted.

(ii) Independent Component Analysis

Independent Component Analysis is actually an extension to the Principal Component Analysis. Actual dimensionality reduction (selection of number of components) is decided with PCA and ICs are used for prioritising the features. In PCA, we get uncorrelated components but in ICA we have independent components. A neural algorithm proposed in [6] is implemented here. The method used in feature prioritisation is same as that used in PCA.

(iii) K-means clustering

K-means clustering technique is used for grouping the total available samples in k clusters. The k is specified in advance before actually implementing the algorithm. Generally clustering is performed by using similarity between the patterns. This method uses distance (either Euclidean or any other) as similarity measure. First the tentative cluster centers are calculated (or specified) and the distance of a pattern is calculated with each cluster center. The pattern belongs to the cluster whose cluster center is most similar to it i.e. the distance is minimum. Once a new pattern is added to a cluster its center is shifted. The algorithm is repeated till no change in classification takes place.

(iv) Fuzzy C-means Clustering

Fuzzy C-means method is an extension to the k-mean clustering. Here each pattern belongs to all C clusters with a membership. The sum of all these membership values is always 1. A pattern is said to be classified to the cluster where its membership is highest. Here also Euclidean distance is used as similarity measure.

(v) Self Organizing Map (SOM)

Kohenon network performs clustering through a competitive learning mechanism called "winner takes all". In essence, the node with largest activation level is declared the winner in the competition. This node is the only node, suppressed to the zero activation level. Furthermore, this node and its neighbours are the only nodes permitted to learn for the current input pattern. After training, the weight vector of each node encodes the information of a group of similar input patterns. Given an input

vector, it is assigned to the node with the maximum activation. Since the number of nodes is fixed, the net algorithm is similar to the K-means clustering algorithm.

(vi) Adaptive resonance theory (ART2)

ART2 is widely used clustering technique for analog or continuous valued patterns. The patterns are classified or clustered with the accuracy defined by a factor called "vigilance factor". Their ability to generalize is limited; however, the ability of this network to create new pattern class in its knowledge base on the arrival of novel pattern makes it very suitable for clustering. The classification is dependent on the presentation of input patterns.

(vii) Fuzzy ART

Fuzzy ART, can classify both binary and analog valued patterns. In this network also, the clustering is mainly dependent on the factor called vigilance factor and order of presentation of input patterns also plays role in classification. This is also an unsupervised network, because we need not to specify number of clusters. This is determined by the network itself depending upon the vigilance factor

(viii) ID3 algorithm

Id3 algorithm is a classification tree approach based on the entropy calculation. It can also be called as inductive learning approach. For using this method for clustering first input set has to be modified. We are required to divide each feature in some ranges. Then a classification tree is built based on minimum entropy approach. At every node a feature having minimum entropy is selected and then a decision is taken there. Classification continues till all the available final nodes are leaf nodes. All the patterns belonging to same leaf node belongs to same group.

1.2 Problem statement

To select the important variables and to cluster the data samples for modelling and prediction of steel making converter life using operating parameters.

The life of steel making converter depends on various features. Here an attempt has been made to categorize the variables and obtain a reduced dimensioned

feature space. Then based on these selected features, clustering of the samples has been done to find out the heterogeneity in the data.

1.3 Organization of thesis

Data prioritisation techniques Principal Component Analysis and Independent Component Analysis is discussed in chapter 2. Chapter 3 contains detailed description of all the clustering algorithms. Results of respective techniques are given along with there discussion. In chapter 4, a comparative study of all these methods is given.

CHAPTER 2

DATA PRIORITISATION TECHNIQUES

Data prioritisation or feature prioritisation is an important aspect of dimensionality reduction. Feature prioritisation is related with the selection of feature which can capture most of the variation available in the total variables. Principal Component Analysis (PCA) and Independent component Analysis (ICA) can perform this task. This chapter contains a detailed description of these two techniques.

2.1 Principal Component Analysis

The central idea of principal component analysis (PCA) is to reduce the dimensionality of a data set which consists of a large number of interrelated variables ,while retaining as much as possible of the variation present in the data set . This can be achieved by two methods . One ,by transforming the variables to a new set of variables called the principal components (PCs) which are uncorrelated , which are ordered so that the first few retain most of the variation present in all of the original variables . Second , by priortising the original variables based on obtained PCs .Then we select the variables according to their preference . These selected variables cover most of the information contained in the total data set.

So, we can say although p (total no. of variables) components are required to produce the total system variability , often much of this variability can be accounted for by a small number k , of the principal components. If so, there is (almost) as much information in k variables as there is in the p variables.

In proposed work the dimensionality reduction is carried out by priortising the variables.

Algebraically, principal components are linear combination of the p random $X_1,X_2,.....X_p$ variables . Geometrically, these linear combinations represent the selection of a new co-ordinate system obtained by rotating the original system with $X_1,X_2,.....X_p$ as the co-ordinate axes. These new axes represent the directions with

maximum variability and provide a simple and more parsimonious description of the covariance structure.

Principal components depend solely on the covariance matrix Σ (or the correlation matrix ρ) of $X_1, X_2, \ldots\ldots X_p$. Their development does not require a multivariate normal assumption. On the other hand, principal components derived for multivariate normal populations have useful interpretations in terms of constant density ellipsoids. Further, inferences can be made from the sample components when the population is multivariate normal.

Let the random vector $\mathbf{X}' = [X_1, X_2, \ldots\ldots X_p]$ have covariance matrix Σ with eigen values $\lambda_1 \geq \lambda_2 \geq \ldots\ldots \lambda_p \geq 0$.

Consider the linear combinations

$$Y_1 = l_1'\mathbf{X} = l_{11}X_1 + l_{21}X_2 + \ldots\ldots l_{p1}X_p$$

$$Y_2 = l_2'\mathbf{X} = l_{12}X_1 + l_{22}X_2 + \ldots\ldots l_{p2}X_p$$

.

.

.

.

$$Y_{p_1} = l_p'\mathbf{X} = l_{1p}X_1 + l_{2p}X_2 + \ldots\ldots l_{pp}X_p$$

The principal components are those uncorrelated linear combinations $Y_1, Y_2, \ldots\ldots Y_p$ whose variances are as large as possible.

The first principal component is the linear combination with maximum variance. That is, it maximizes $\text{Var}(Y_1) = l_1' \Sigma l_1$. It is clear that $\text{Var}(Y_1) = l_1' \Sigma l_1$ can be increased by multiplying any l_1 by some constant. To eliminate this indeterminacy ,it is convenient to restrict attention to coefficient vectors of unit length. We therefore define

First principal component = linear combination of $l_1'\mathbf{X}$ that maximizes

$$\text{Var}(l_1'\mathbf{X}) \text{ subject to } l_1'l_1 = 1$$

Second principal component =linear combination of $l_2'\mathbf{X}$ that maximizes $\mathrm{Var}(l_2'\mathbf{X})$

subject to $l_2'l_2 = 1$ and $\mathrm{Cov}(l_1'\mathbf{X}, l_2'\mathbf{X}) = 0$

i^{th} principal component =linear combination of $l_i'\mathbf{X}$ that maximizes

$\mathrm{Var}(l_i'\mathbf{X})$ subject to $l_i'l_i = 1$ and $\mathrm{Cov}(l_i'\mathbf{X}, l_k'\mathbf{X}) = 0$

for $k<i$

Principal components can be evaluated by performing eigen value - eigen vector analysis on covariance matrix Σ (or correlation matrix ρ when variables are of different dimensions). Let Σ have the eigenvalue -eigenvector pairs $(\lambda_1, e_1), (\lambda_2, e_2), \ldots\ldots (\lambda_p, e_p)$ where $\lambda_1 \geq \lambda_2 \geq \ldots\ldots\ldots\ldots \geq \lambda_p \geq 0$. The i^{th} principal component is given by

$$\mathbf{Y_i} = \mathbf{e_i'X} = e_{1i}X_1 + e_{2i}X_2 + \ldots\ldots\ldots\ldots e_{pi}X_p \quad , \quad i = 1,2,\ldots\ldots\ldots\ldots p$$

With these choices ,

$$\mathrm{Var}(Y_i) = e_i'\Sigma e_i = \lambda_i \quad i = 1,2,\ldots\ldots\ldots p$$

$$\mathrm{Cov}(Y_i Y_k) = e_i'\Sigma e_k = 0 \quad i \neq k$$

If some λ_i are equal, the choice of corresponding coefficient vectors $\mathbf{e_i}$, and hence $\mathbf{Y_i}$, are not unique .

Therefore, the principal components are uncorrelated and have variances equal to the eigenvalues of Σ.Hence

$$\text{total sample variance} = \lambda_1 + \lambda_2 + \ldots\ldots\ldots\ldots + \lambda_p$$

consequently, the proportion of total variance due to k^{th} principal component is

$$\begin{pmatrix} \text{proportion of total} \\ \text{population variance} \\ \text{due to } k^{th} \text{ principal} \\ \text{component} \end{pmatrix} = \frac{\lambda_k}{\lambda_1 + \lambda_2 + \ldots\ldots\ldots\ldots + \lambda_p} \qquad k=1,2\ldots\ldots\ldots\ldots,p$$

If most (for instance, 80 to 90%) of the total population variance, for large p, can be attributed to the first one, two or three components, then these components can replace the original p variables without much loss of information.

Each component of the coefficient vector $e_i' = \left[e_{1i}, e_{2i}, \ldots\ldots\ldots\ldots, e_{pi}\right]$ also merits inspection. The magnitude of e_{ki} measures the importance of k^{th} variable to the i^{th} principal component. In particular, e_{ki} is proportional to the correlation coefficient between Y_i and X_k .

In proposed work, the variables have been prioritised and depending upon the sample variances small number of variables have been selected for modelling purpose. The results are then compared with the variables selected by the R & D persons.

2.1.1 Selecting a subset of Variables

When p, the number of observed, is large it is often the case that a subset of m variables with m<p, will contain virtually all the information available in all p variables. It would be useful to determine an appropriate value of m, and to determine which subset of m variables is best.

Solution of these two problems, the choice of m and the selection of a good subset, depends on the purpose to which the subset of variables is to be put. If the purpose is simply to preserve most of variation (as in our problem) then the PCs can be used fairly straightforward to solve both problems.

Regarding the choice of m, the following methods have been proposed

(i)the total sample variation with m variables is calculated, which is the measure of variation contained in m variables. If it is about 80-90% then we can go for these small number of variables.

(ii)the PCA is performed on correlation matrix ρ rather than covariance matrix Σ .Then the those many components can be chosen for which $\lambda \geq 1$.

pc1	0.3087	0.3514	0.4838	0.4601	-0 0884	0.3058	0 3661	0 6326	0 183	0 302	0 2473	0 0713	0 036	0 3996	0 9332	0 3053	0 1913	0.4381	0.4002	0.1994	0 2985	0.9855	0.9889	0 998	1	0.9906
pc2	0.2118	0.5847	0 848	0 8876	-0.7867	1	0.2681	-0.053	0 2822	0 1042	0 2209	-0 0641	0.019	-0 076	-0 2399	0 3478	0.0066	0 2869	0 0007	-0 5834	-0 0613	-0 3048	-0 306	-0 301	-0.3019	-0 306
pc3	-2.6446	-1.9228	0 8516	1	-2 3443	0 2818	-1 8669	0.7634	-1.912	-1 552	-2 1611	-1 2036	-0.7644	0 5074	0 654	-1 9601	-1.308	-0 2881	0 2062	-2.4059	-1 6075	0 6424	0 6176	0 6221	0 6141	0 5589
pc4	-0.2265	-0.0416	0.0771	0.0776	-0.1496	0.0352	-0.1771	-0.324	-0 492	-0.061	-0 272	0 0831	-0.5655	0 2268	-0 0333	-0 3571	0 8569	0 9644	-0 0801	0 306	1	-0 0485	-0 116	0 08	-0.0793	-0.096
pc5	0 1121	0.3472	-0 0636	0.011	0.2738	-0 01	0.3393	0.4653	-1 084	1	0 1018	-0 8369	-1.1806	0 7064	0.0339	0 4806	-0 437	-0 3923	-0 5724	0.0449	-0 0844	-0 1009	-0 113	-0 083	-0 088	-0.128
pc6	-0 2761	0 2687	0 0858	-0.0418	0.2158	0.0267	0.6041	0.8169	0.3174	-0 263	-0 5677	-1 9101	0.482	-0 764	0.0056	-0 263	1	-0.004	0 0617	0 457	-0.1735	-0 0518	-0 084	-0 076	-0 0951	-0.046
pc7	-0.659	0.9851	0 2196	0.3927	0 3753	0 6295	0.3626	-0.016	0.3253	-1.022	-1 1162	1	-0.6451	-0 792	-0 2029	-0 0243	-0.192	-0 9596	-2.3603	0 729	0.2495	0 2853	0 2868	0 3014	0.2791	0 3192
pc8	0 086	-0 658	0.0241	-0.0167	-0.5221	-0.109	-0 3169	0 0252	0 3538	1	0 9462	-0 1734	-0 2995	-1 051	-0 1277	-0 6704	0 2211	-0 053	-0 857	-0 1884	0 1181	0 1599	0 1609	0 1527	0.142	0.1704
pc9	-0.0092	0.1841	-0.0839	-0.1068	0 1536	0 0406	0.3093	-0 1	0 0861	0 0304	-0 3029	0 2025	-1 1474	-1 112	0 0554	0 0157	-0 334	0 1363	1	-0 0585	-0 1185	0 0117	0 0277	0 0272	0 0015	0.0271
pc10	-0 2056	-0 0983	0.1381	-0.0272	-0 1357	-0.115	1	0 2443	-0 289	0 8883	-0 6191	0 7743	0.4408	0 0753	0 0265	-0 6209	0 2939	-0 1537	0 1108	-0 1074	-0 2241	-0 0796	-0 086	-0 079	-0 0726	-0 084
pc11	-1.0789	1	0.4716	0 0758	0 6365	0.2621	-1.0752	-0.105	0.1284	0 6401	0 4516	0 1578	0 023	0 1649	0.1347	-0 5562	0 3457	-0 4487	0 515	0 3951	-0 3548	-0 0582	-0 062	-0 058	-0.052	-0.027
pc12	-0.7019	-0.2408	-0.0137	0 1813	-0 0098	-0.201	-0.4092	1	-0 095	0 3406	-0 1534	0 3086	0 411	-0 488	-0.04	0 9223	-0 38	0.3071	0.0174	0.1154	0.5991	-0.1439	-0.118	0 11	-0.1354	-0.1
pc13	0.6653	0 3102	0 2454	0 0815	0.1147	0.2505	-0 5655	-1.2	-0 984	0 7064	-0.8694	-0 4914	1	-0 788	-0 0444	0 1048	-0 602	-0 009	0 0134	0 2277	0.3323	0 2221	0 1688	0 1541	0 181	0 1845
pc14	0.3256	-0.2489	-0.0319	-0.1232	-0.08	-0 077	-0 66	-0 161	0 5908	0 507	-0 9272	0 1055	-0.3604	0 1617	0 1094	0 8419	1	-0 357	-0.0728	-0.4499	-0 3475	0 0811	0.0323	0 0011	0 0048	0 0388
pc15	-0.3619	-0 0811	-0 0646	-0.0162	-0 1316	0 0597	0 2708	-0.303	-1 279	-0 463	0 7765	0 3269	0.1836	-0 601	0.113	0.813	1	-0 2448	0 0033	-0.0726	-0.6556	0.0926	0.0581	0 0948	0.0924	0 0694
pc16	-1.7803	-0.5623	-0.1564	-0.1336	-0 4188	0.0928	1	-1 61	0.8429	0 5266	0 056	-0 6317	-0 008	0 4173	-0 0733	0 8466	-0 405	0 0921	0 0257	0 3984	0.103	0.1571	0 1867	0 099	0.1421	0.1708
pc17	-0.4107	0.8227	-0 2807	-0 5432	0 4803	-0 246	0.2657	-0.11	0 0018	-0 059	0 149	0 1817	0 2244	0 09	-0 1741	-0 0368	0 1417	-0 2122	-0 0731	-1.9307	1	0 2393	0.103	0 1147	0 132	0.107
pc18	-0.1018	0.4782	-0.6256	0.1823	0.093	-0.187	-0 0849	-0 01	0 0573	0 1715	-0 0927	0 0342	0.0783	-0 022	-0.2357	-0 0017	-0 073	1	-0 3378	-0.1607	-0.7641	0 002	0 0753	0 0845	0.1038	0.0774
pc19	0 0166	0 1148	-1.2476	1	-0 3397	0.1767	-0 0512	0 0224	-0 029	0 0794	0.0022	-0 0261	0 033	0 0632	-0 2652	-0.1457	0.1749	-0.6133	0.362	0 1279	0 3546	0 1069	0 0978	0 0494	-0 0084	-0.018
pc20	0.0621	-0 1241	1	0 0349	0.0387	-0.63	-0.0003	0 2749	-0 156	-0 048	-0.0192	0 0132	-0 1786	0 3132	-3.3693	0.1523	0.1885	-0 1841	0.5396	0.2589	-0 089	0.5065	0.5188	0 4576	0.4824	0 4292
pc21	-0 0214	0 2449	-0 4714	-1.1794	-0 6916	1	-0.146	0 3145	-0 073	0 0751	-0 047	0 0659	0 0651	0 0828	-0 2547	0.0031	-0 031	0 0149	0 0843	0 2088	0 0109	0 0951	0 0225	0 0074	0 0585	0.1178
pc22	-0 0342	-0 7365	-0 1291	0 1798	1	0 0023	0 0474	0 0546	-0 03	0 0687	0 0076	0 0111	0 0276	0 027	-0.1806	-0.0024	0 0437	0.1706	-0 0193	-0 2333	-0.0686	0 0007	0.0104	0 026	0.0191	0.0671
pc23	-0.0097	-0 0104	0 0237	-0 0179	0 0364	0 0193	0.0008	0 0242	0 0132	0 001	0 0136	0 017	0 005	-0 013	0 0195	0 0021	-0 031	0 0616	-0 0241	0.0294	-0 0503	1	0.2366	-0 279	-0.1556	-0.83
pc24	-0.0015	0 0115	0 0132	-0.0485	-0.0032	0 0267	-0 0065	-0 003	-0 017	0.0013	-0 0081	-0 0055	0 0041	-0 002	0 0089	-0 0014	0 0251	-0 0011	-0 0058	-0 0148	0.0233	-0 5359	1	0 0301	-0.1555	-0 345
pc25	-0.0096	-0 0101	-0 0062	-0.0381	-0 0096	0.0367	-0 0116	-0 01	0 0301	0 0129	-0 0152	-0 0086	0 0109	-0 009	-0 0083	0 0089	0.0055	-0 0172	0 0072	0 0057	0.01	-0 211	-0.36	1	0 3448	-0.758
pc26	0 0018	0 0155	0 0206	-0.0276	-0 0024	-0.013	0 0077	-0 015	-0 013	0 0013	0 0095	0.0064	0 0115	0 0277	-0 0054	-0 0023	-0 017	0 0405	-0 0155	-0 0022	-0 0328	0 3407	0.0158	1	-1.758	0 4153

table 2.1.1 : principal compomemts for campaign no. 1\2

S. No.	Comp loading	Cum loading
1	0 2718	0 2718
2	0 1417	0 4135
3	0 1013	0 5148
4	0 0724	0 5872
5	0 0421	0 6353
6	0 0397	0 675
7	0 0381	0 7131
8	0 0352	0 7483
9	0 034	0 7823
10	0 0298	0 8121
11	0 0286	0 8407
12	0 0243	0 8651
13	0 0226	0 8877
14	0 0211	0 9087
15	0 0195	0 9282
16	0 0182	0 9464
17	0 0151	0 9615
18	0 0119	0 9734
19	0 0073	0 9807
20	0 007	0 9877
21	0 0054	0 9931
22	0 004	0 997
23	0 0013	0 9984
24	0 0009	0 9992
25	0 0006	0 9998
26	0 0002	1

table 2 1 2 : componet loading for campaign no 1\2

Moving on to the choice of m variables, following method is used

Associate one variable with each of first m PCs, namely the variable not already chosen, with highest coefficient(value of e_{ki}), in absolute value, in each successive PC. These m variables are retained, and the remaining $m^* = p-m$ are deleted.

2.1.2 Result of PCA

PCA has been performed on all the data sets of 15 campaigns, but results of only two campaigns (1\2 and 1\4) are presented as all the results are almost similar. As given in table 2.1.2 for campaign no. 1\2, the cumulative loading for 14 components is 90.87% i.e. 14 components capture 90.87% of the total variation present in data. Similarly, in table 2.1.4 for campaign no. 1\4, the cumulative loading for 14 components is 90.72%. Therefore, we see that 14 components cover almost 90% of the total information present in data sets. The variables are selected using aforesaid criterion. With reference to table 2.1.1, following variables are chosen for campaign no. 1\2

1. % CaO in slag
2. % P in Hot metal
3. Hot metal weight
4. Average Lime addition
5. Tap-Tap time
6. Tapping Temperature
7. Ore addition
8. Hot metal Temperature
9. Bath Carbon
10. Scrap weight
11. Hot metal Carbon
12. Dolo addition
13. % Si in hot metal
14. % FeO in slag

For campaign 1\4, following variables are chosen using table 2.1.3

1.% CaO in slag
2. Hot metal weight

pc1	0.0989	0.0952	0.3472	0.135	-0 0004	0.0089	0 1175	0.6255	0.0406	0.7908	0.7643	0.0499	0.0096	0.4489	0.9608	0 0799	0.238	0.4433	0.7253	0.4295	0 4107	0.999	0 9978	0 9995	1	0 9962
pc2	1	0.2441	0.1028	0.3029	-0.2143	0.2723	0.8453	-0.1619	0.5608	-0 2369	-0.1926	0.2394	0 265	0.0251	-0.106	0 7759	0.7567	0 7168	-0 0959	0.4989	0 7297	-0 1655	-0.1651	-0 1577	-0 1633	-0 1649
pc3	-0 6772	-1.1356	0.0064	-0.819	-0.0843	0.8747	-0.8459	-0.1706	-0.7592	-0.5465	-0.6075	-0.0635	-0.6081	0.3279	0.0635	-0 3803	0.82	0 8863	-0.1433	0.5963	1	0.0167	0.017	0.0185	0 0189	0.0183
pc4	0 0842	-0.3328	0.7349	0.0476	-1.0171	1	0.0728	0.1744	0.1039	-0.224	-0.2426	-0.0649	0.1084	0.1253	0.1287	0 3351	-0.2529	-0.1768	0 0557	-0.6616	-0 446	0.0756	0.0738	0 0669	0 0721	0.0664
pc5	0.7401	-1.2684	-0.348	-1.781	1	-0.064	0.3252	0.6152	0.9993	-0.2769	-0.3298	0.2019	0.7783	0.1708	0.1737	0.2621	-0.1902	-0.7195	-0.3271	-0.1083	-0 153	0.1713	0.1851	0.1594	0.1659	0 1653
pc6	0.2808	0.1154	-0.615	-0.113	0.3491	0.0461	0.9668	0.2802	-0.9326	-0.2215	-0.2931	-1 4772	-1 2288	0.6009	0 0811	1	-0.6056	-0.0654	-0.2827	-0.104	-0 015	0.0835	0.0676	0 1002	0.0862	0 0878
pc7	0.1834	-0.061	-0.484	-0.052	-0.1472	0 0467	0.0158	-0.3082	0.0159	0.081	0.1076	1	-0 6946	0.3748	-0.027	0.136	-0.0764	-0.0661	-0.0062	-0.2711	-0.105	0.0374	0.0468	0 0526	0 0429	0 042
pc8	-0.7511	0.8571	1	-0.184	0.7084	0.3634	0 3272	0 6002	-0.1297	-0.4531	-0.5123	0 9413	-0.4725	-0.436	0.098	0.4359	-0.3017	-0 569	-0.2026	0.6979	0 1504	0.0326	0.0406	0.0223	0 0235	0.0236
pc9	-0.2555	0.0151	-0.706	0.3885	-0.3581	0.1798	-0 2495	0.8708	1	-0.0976	-0.0673	-0 2241	-0.5949	-0 573	-9E-04	-0 1658	0.0789	0.0628	-0 568	0 0635	0.0843	0.0793	0.0897	0 0816	0 0719	0.0574
pc10	-0 2606	-0.5675	0.0471	-0.806	0.0757	-0.171	0.4977	-0.1536	-0.4432	0 7231	0.4225	0.0179	-0.7019	-2.484	0.0688	0.9533	1	0 2991	0.9343	-0 8928	-0 282	-0.0494	-0.0565	-0 0555	-0 0478	-0 0266
pc11	0.2777	-0.1942	0.8492	-0.043	0.4101	-0.043	0.8502	0.5659	0.6296	0.057	0.077	-0.2296	-1.1037	0.5643	-0 087	-1.4517	0.3842	0.036	1	-0.3752	0.0014	-0.3168	-0.3266	-0.3075	-0 3202	-0 3163
pc12	0.7213	-0.2133	-0.271	0.0975	-0.3156	0.2075	1	-0.2963	-0.7184	-0.3134	-0 4121	0.3149	0.1407	-1 062	0.1365	-1 3875	-0 8284	0.2167	-0.3779	0.3	0 0958	0.2608	0.2608	0.2713	0 2637	0.2797
pc13	-0.6161	-0.087	-0.605	-0.347	-0.077	0.972	0.5652	0.8123	-0.2486	0.8641	1	0.3059	0 5807	0.1777	-0.154	0.0509	-0.9451	0.5363	0.2506	0 2565	0.4183	-0.4302	-0.4202	-0 4086	-0 4275	-0 4451
pc14	-0 4386	0.4155	-0.44	-0.061	0.024	0.1015	0.5697	0.4372	-0.444	-0.1524	-0.1891	0.1905	0.4831	0.3146	0.049	-0.2811	1	0 0635	-0 337	-0.4958	-0.48	0.0537	0.0601	0 066	0 0593	0 0764
pc15	-0.3489	0.2177	-0.288	-0 031	0.0176	-0.571	-0.0847	0.2856	0.2144	-0.7154	-0 775	0.2097	0.2766	0.0211	0.0494	0.201	-0.6685	1	0.9302	-0.6257	0.1781	0.0427	0.0436	0 0634	0 0524	0 0526
pc16	-0 0188	-1.0451	0.1538	0.424	-0 9009	-1.224	0.2212	1	-0.6128	0.0257	-0.2163	0.4171	0.0224	0.1618	-0.073	0 3043	0.128	-0.3173	0.1329	0.7762	-0.23	-0.1573	-0.1491	-0.1439	-0 1502	-0.1541
pc17	-0.1566	0.3093	-0.431	-0.474	-0.4079	0.2358	0.1241	-0.3522	0.297	-0 2348	-0.0788	-0.2068	-0 0597	0.0234	-0.03	-0.038	0.1051	-0 0625	0.7307	1	-0.88	0.0154	0.0161	0 0053	0 0102	0.0275
pc18	0.5584	-0.446	-0.128	0.8889	1	0.6143	-0.4901	0.3746	-0.3031	-0.1965	0.0078	0.1836	0.0396	-0.136	0.1392	0.1334	0.0503	0.4223	0.2441	0.1954	-0 906	-0 0566	-0.0566	-0 0406	-0 0584	-0 0777
pc19	1	0.8273	-0.009	-0.699	-0.3531	-0.028	-0 7831	0.7221	-0.4939	0 0525	0.0072	0.0271	-0.104	-0 11	0.1022	-0.1487	0 0711	0 015	0.078	-0.0649	0 0584	-0.12	-0.1213	-0 1155	-0 1136	-0.1107
pc20	-0.0591	0.0265	0.532	-0.4	-0 0212	-0.348	0.048	-0.0376	0.1603	0 2287	0.1656	0.0289	-0.2003	0 0947	-0.173	0.0124	-0.1946	1	-0.7555	0 1066	-0 693	0.0212	0.0209	0 0137	0 0194	0.0177
pc21	0.0749	-0.0424	0.1018	-0.047	0.0044	0.0157	-0.0023	0.231	-0.163	-0.9885	1	0.0022	0.0098	-0.061	-1.529	0.0339	0.05	-0 0475	0.1465	-0.0624	0.1048	0.2539	0.2589	0.2376	0 2479	0 2305
pc22	0.083	0.0371	-0.027	0.0608	0.0936	0.1981	-0.067	0.0955	-0.0025	1	-0 9331	-0.0091	0.0438	0.0526	-1.104	-0.0066	0 0099	-0.0308	0.1678	0.0436	0.0001	0 1467	0.1585	0 184	0.1498	0 1285
pc23	-0.0012	0.017	0.0019	-0.027	-0.0141	-0.024	0.0179	-0.0313	-0.0221	-0.0183	0.0034	-0.0102	0.0049	-0.006	0.0428	-0 0103	0.0187	-0.0023	0.0083	-0.0069	-0.006	0.6495	1	0.6971	0 0077	-2.3694
pc24	-0.0057	0.0001	0.0095	-0.009	-0.0034	-0.001	-0.004	-0.0055	0.0086	-0 0124	0.0134	-0.0016	0.004	-0 006	0.0043	0.001	0 0023	-0 0085	-0 0071	0.0052	-0 003	-0.3913	-0.5846	1	0 0413	-0.0593
pc25	-0.0027	0	-0.008	-0.001	-0.0005	0.0015	0.0024	0.0052	0.0025	0 0041	-0.0062	0 0113	0.0011	0 0004	-0.005	-0.0015	0.0016	-0.0013	-0 0013	0.0021	-0.002	1	-1.1003	-0.2959	0 6759	-0 2754
pc26	0 0012	-0.0013	0.0001	-4E-04	-0.0022	-0 003	-0.0038	0.0007	-0.0018	-0.0001	0.0022	0 003	0.0024	0.0019	-0.001	0.0023	-0.0008	0.0015	-0.0008	-0 002	0.0027	1	-0.1788	0 3661	-1 4874	0 3016

table 2.1.3 : principal components for campaign no. 1\4

S. No.	Comp. loading	Cum. loading
1	0.3267	0.3267
2	0.0854	0.4121
3	0.0849	0.495
4	0.0694	0.5644
5	0.063	0.6273
6	0.0462	0.6735
7	0.0376	0.7112
8	0.0354	0.7466
9	0 0323	0.7789
10	0.0318	0.8106
11	0.0273	0.838
12	0.0247	0.8627
13	0 0236	0.8863
14	0.0209	0.9072
15	0 0193	0.9265
16	0.0167	0.9432
17	0.0143	0.9575
18	0.0135	0.971
19	0.0102	0.9812
20	0.0093	0.9905
21	0.0045	0.9951
22	0.0043	0.9994
23	0.0003	0.9997
24	0.0002	0.9999
25	0.0001	1
26	0	1

table 2.1.4 : component loading for campaign no. 1\4

S. No.	Variable
1	hm weight
2	Si in hm
3	Mn in hm
4	blow oxygen
5	Tap temperature
6	Tap-tap time
7	Lime
8	Ore
9	Bath Carbon
10	Sulpher
11	Phosphorus
12	Basicity
13	FeO in Slag
14	CaO in Slag

table 2.1.5 : Variables Selected through PCA

3. Bath Phosphorus
4. Bath Sulphur
5. Mn in Hot metal
6. Tap-tap time
7. Average Lime addition
8. %Si in Hot metal
9. Blow Oxygen
10. Ore addition
11. Lance Ht.
12. Tapping Temperature
13. Slag Basicity
14. %FeO in slag

Similarly, for all campaigns PCA has been performed and common variables are finally selected. The list of selected variables is given in table 2.1.5.

2.2 Independent Component Analysis

Independent component analysis (ICA) is an extension of Principal component analysis (PCA), that has been developed in context with blind separation of independent sources from their linear mixtures. In a sense, the starting point of ICA is the uncorrelatedness property of standard PCA. Roughly speaking, rather than requiring that the coefficients of a linear expansion of the data vectors be uncorrelated, In ICA they must be mutually independent (or as independent possible).

In proposed work, neural network implementation of ICA as suggested in [5] is implemented.

The ICA network structure

Let us denote $\mathbf{x}_k = [x_k(1), x_k(2), \ldots\ldots\ldots\ldots x_k(L)]^T$ the L-dimensional k^{th} data set, where k = 1,2,..... L = no. of features or variables, M = no. of independent components.

Suppose we write a signal model,

$$\mathbf{x}_k = \mathbf{A}\mathbf{s}_k + \mathbf{n}_k = \sum_{i=1}^{M} s_k(i)a[i] + \mathbf{n}_k$$

Here $\mathbf{s}_k = [s_k(1), s_k(2), \ldots\ldots\ldots s_k(M)]^T$ is the source vector consisting of M source signals (independent components) $s_k(i)$ (i= 1,2,.....M) at the index value k. $\mathbf{A} = [\mathbf{a}(1), \mathbf{a}(2), \ldots\ldots\ldots \mathbf{a}(m)]$ is a constant $L \times M$ " mixing matrix " whose columns a(i) are the basis vectors of ICA, and $\mathbf{n}_k$ possible corrupting additive noise.

Now, let us define the estimated expansion by

$$\mathbf{x}_k = \mathbf{Q}\mathbf{y}_k + \mathbf{n}_k'$$

where $\mathbf{y}_k$ is the estimate of original independent signal s_k.

or suppose

$$\mathbf{y}_k = \mathbf{B}_k\mathbf{x}_k$$

where $\mathbf{B}_k$ is an $M \times L$ matrix called separating matrix.

Here, the $L \times M$ matrix **Q** denotes the estimate of the ICA basis matrix **A**, $\mathbf{y}_k$ is the estimate of the independent component vector $\mathbf{s}_k$ and $\mathbf{n}_k'$ is the noise or error term. The first task is separation of sources or estimation of the vector $\mathbf{y}_k$. This can be done by learning separating matrix **B** using some suitable algorithm. After this the components of the vector $\mathbf{y}_k$ should be as independent as possible. For learning the matrix **Q**, we then simply minimise the mean square error $E\left\{\left\|\mathbf{n}_k'\right\|^2\right\} = E\left\{\left\|\mathbf{x}_k - \mathbf{Q}\mathbf{y}_k\right\|^2\right\}$ with respect to **Q**.

This estimation procedure can be realised using the two layer feed forward network.

2.2.1 Algorithm for ICA

The algorithm for ICA consists of three steps:

(1) Whitening of the data set.

(2) Separating algorithm.

(3) Estimation of the Basis Vectors of ICA

Whitening:

Prior to inputting the data vectors $\mathbf{x}_k$ to the ICA networks, they are made zero mean by subtracting the mean, if necessary. This normalises the data with respect to the first order statistics. Next, the data vectors are made uncorrelated using Principal

Component Analysis (PCA). PCA is often used for whitening, because then one can simultaneously reduce the dimensions also. The PCA whitening matrix is given by

$$\mathbf{V} = \mathbf{D}^{-1/2}\mathbf{E}^{T}$$

Here the $M \times M$ diagonal matrix $\mathbf{D} = \text{diag}[\lambda(1), \lambda(2), \ldots\ldots\lambda(M)]$, and $L \times M$ matrix $\mathbf{E} = [c(1), c(2), \ldots\ldots\ldots\ldots, c(M)]$, with $\lambda(i)$ denoting the i^{th} largest eigenvalue of the data correlation matrix, and $c(i)$ the respective i^{th} principal eigenvector.

Now, the whitened vectors are given by

$$\mathbf{v}_k = \mathbf{V}\mathbf{x}_k$$

Separating Algorithm

The core part in ICA is learning of the separating matrix **B**, which can be written as

$$\mathbf{B} = \mathbf{W}^{T}\mathbf{V}$$

where, $\mathbf{W}^{T}$ is the orthogonal $M \times M$ separating matrix applied to whitened vectors $\mathbf{v}_k$.

Learning rule:

$$\mathbf{W}_{new} = \mathbf{W}_{old} + \mu_{old}\left[\mathbf{v}_{old} - \mathbf{W}_{old}\,g(\mathbf{y}_{old})\right]g(\mathbf{y}_{old})^{T}$$

where,

μ_k = Learning rate

$g(t) = \tanh(t)$

Estimation of the Basis Vectors of ICA

After calculating the separating matrix **B**, the basis vectors can be calculated easily using following relation:

$$\mathbf{A} = \mathbf{B}^{T}\left(\mathbf{B}\mathbf{B}^{T}\right)^{-1} = \mathbf{E}\mathbf{D}^{1/2}\mathbf{W}$$

After estimating these basis vectors, we can use them for selecting the variables as in PCA, i.e. Associate one variable with each of basis vectors, namely the variable not already chosen with highest coefficient, in absolute value in each successive basis vector. These M variables are retained and remaining variables are deleted.

2.2.2 Results of ICA

As PCA, ICA has also been performed for all 15 campaigns, but results of only 2 campaigns (1\2 and 1\4) are presented here. From table 2.1.2 and 2.14, we see that 13

ic1	ic2	ic3	ic4	ic5	ic6	ic7	ic8	ic9	ic10	ic11	ic12	ic13
-0.3058	0.2783	-0.0225	0.0136	0.0492	-0.2935	-0.1927	-0.4875	-0.1274	0.0697	-0.4761	-0.4724	-0.334
0.0808	0.5207	0.6608	0.5263	0.1732	0.4144	0.4218	0.9312	0.0382	1.0951	0.3828	0.5545	0.0972
-0.235	0.0403	0.1776	0.0225	-0.1233	-0.0917	-0.0372	0.3524	-0.225	0.3646	-0.0617	0.0829	-0.3
-0.1174	0.1653	0.1117	0.2063	-0.037	-0.0712	-0.0304	0.1551	-0.1357	0.2228	-0.0786	0.0122	-0.2089
0.0218	-0.057	0.0666	-0.0023	-0.0763	0.1496	0.0995	0.2511	-0.0445	0.1869	0.1549	0.1726	0.0959
-0.0047	0.4367	0.411	0.4028	0.1383	0.1819	0.2136	0.4608	0.0004	0.6948	0.1221	0.2332	-0.0295
0.3122	0.8837	0.3953	0.767	0.5662	0.0961	2.1941	-0.1582	0.4519	0.3852	0.0536	-0.0495	0.0994
-0.5716	-0.45	-0.4223	-0.4409	-0.5621	-0.5287	-0.4917	-0.5127	-0.591	-0.4476	-0.53	-0.6441	-0.6417
0.0822	0.6753	0.7396	0.4724	0.3137	0.5416	0.5616	0.639	0.1323	1.1696	0.3087	0.3167	0.2488
-0.4044	-0.619	-0.0726	-0.5925	-0.3288	-0.3138	-0.2172	0.3698	-0.3289	-0.2017	-0.1364	0.1399	-0.421
-0.4832	-0.2539	0.2175	-0.4911	-0.2381	-0.0078	0.0592	0.452	-0.3862	0.4807	-0.1487	0.0647	-0.2657
1.4012	1.0656	0.7133	1.7355	1.1233	0.9115	0.8856	1.1064	1.1747	0.3448	1.1006	1.4882	1.1365
-0.0859	-0.0813	0.0544	-0.2405	-0.0474	0.0815	0.0727	-0.0195	-0.0757	0.0998	-0.034	-0.1662	0.0144
-0.2107	-0.4318	-0.4655	-0.3878	-0.2579	-0.4996	-0.4688	-0.3176	-0.2386	-0.6421	-0.3846	-0.2493	-0.388
-0.9801	-0.755	-0.7469	-0.7351	-0.8849	-1.0465	-0.9569	-0.6728	-0.961	-0.7163	-0.9602	-0.8207	-1.1686
0.2067	0.4267	0.1666	0.5278	0.2219	0.2481	0.2469	-0.0469	0.1375	0.1378	0.0827	0.0127	0.2214
-0.1434	-0.6937	0.555	-1.6035	0.2107	0.097	0.3047	0.4883	0.3383	0.2818	0.3166	0.0613	0.0415
0.5015	0.2261	0.2389	0.226	0.4909	0.149	0.2306	0.14	0.5667	-0.1477	0.3415	0.2919	0.302
-0.0379	-0.1533	0.3916	-0.3674	0.1114	0.1074	0.227	0.4915	0.1294	0.304	0.2394	0.2809	0.002
0.0709	0.0172	0.0337	0.0975	-0.0261	0.0462	0.0211	0.1588	0.0044	0.0744	0.1128	0.1102	0.0156
0.6689	0.2928	0.1215	0.5332	0.4861	0.2321	0.2402	0.147	0.5914	-0.2793	0.4202	0.4095	0.443
-1.1696	-0.8138	-0.9539	-0.7642	-1.058	-1.2592	-1.1756	-0.9343	-1.1666	-0.8725	-1.2108	-1.0544	-1.3858
-1.1808	-0.8033	-0.9474	-0.7503	-1.0638	-1.2519	-1.1687	-0.9301	-1.1805	-0.8537	-1.2172	-1.0563	-1.3877
-1.1649	-0.8001	-0.9457	-0.7404	-1.0542	-1.2498	-1.1662	-0.9234	-1.1666	-0.8622	-1.2057	-1.0445	-1.3806
-1.1731	-0.81	-0.9533	-0.7531	-1.0602	-1.2603	-1.1758	-0.9273	-1.1733	-0.8701	-1.2151	-1.0493	-1.3901
-1.167	-0.7952	-0.9271	-0.741	-1.0547	-1.2319	-1.1501	-0.8984	-1.1692	-0.8288	-1.193	-1.0334	-1.3728

table 2.2.1 : Independent Components for campaign no. 1\2

components cover almost 88% of the total information. These 13 components for campaign no. 1\2 and 1\4 are given in tables 2.2.1 and 2.2.2 respectively. These 13 components are dominated by some variables in their respective direction. Variables for each component is selected using aforesaid criterion. From table 2.2.1 following variables are chosen for campaign no. 1\2

1. Hot metal temperature
2. Blow Oxygen
3. Tap-Tap time
4. Average Lime addition
5. Ore addition
6. Dolo addition
7. Bath Carbon
8. %MnO in slag
9. %CaO in slag
10. %MgO in slag
11. %FeO in slag
12. Basicity in slag
13. Phosphorus in Hot metal

Using table 2.2.2, following variables are chosen for campaign no. 1\4 following variables are chosen

1. Phosphorus in Hot metal
2. Sulphur in Hot metal
3. Hot metal temperature
4. Blow oxygen
5. Tap-tap time
6. Average Lime addition
7. Ore addition
8. Bath Carbon
9. %MnO in slag
10. %CaO in slag

ic1	ic2	ic3	ic4	ic5	ic6	ic7	ic8	ic9	ic10	ic11	ic12	ic13
0.7735	0.7008	1.0535	0.7536	0.8956	0.6643	0.8399	0.8482	1.1188	0.9187	1.1163	0.8405	0.8137
0.1581	0.1248	-0.1549	0.1634	0.4562	-0.0157	0.1931	0.2014	0.0789	0.2783	-0.1371	0.119	0.154
-0.4515	-0.586	-0.3469	-0.4459	-0.0609	-0.82	-0.2677	-0.1821	0.2379	0.0664	-0.1314	-0.4676	-0.2956
-0.0831	-0.1873	0.0945	-0.0992	-0.0107	-0.2462	0.0129	0.0257	0.2549	0.1229	0.1458	-0.0377	-0.0274
0.651	0.5966	0.3807	0.6581	0.5168	0.5858	0.6854	0.6545	0.4468	0.6656	0.2887	0.6441	0.6332
-0.7742	-0.6789	-1.0203	-0.7677	-0.3933	-0.7791	-0.8491	-0.8052	-0.9281	-0.8063	-0.9445	-0.8724	-0.8006
0.2936	0.2278	0.0857	0.2757	0.6149	0.0872	2.3413	0.353	0.3031	0.4409	0.1114	0.2686	0.288
0.0623	-0.4043	-0.1781	0.0032	0.5639	-0.8297	0.4465	0.5171	0.8474	0.9665	-0.0463	0.0657	0.2338
1.6213	1.6133	1.7746	1.6171	1.7245	1.5919	1.642	1.6494	1.799	1.6797	1.828	1.6566	1.6422
-0.3283	-1.0055	-0.6147	-0.4337	0.2597	-1.5261	0.1791	0.2324	0.6129	0.7988	-0.5244	-0.2753	-0.1455
-0.2345	-0.8999	-0.5309	-0.3414	0.3354	-1.405	0.2568	0.3044	0.6514	0.8516	-0.4555	-0.1813	-0.0653
0.5547	0.7089	-0.8308	0.5792	1.8587	0.2924	0.3902	0.4171	-0.4219	0.4599	-0.7885	0.308	0.3838
0.1083	-0.0262	0.8348	0.0765	-0.5692	0.1747	0.2206	0.2053	0.6545	0.2072	0.7975	0.2467	0.1956
-0.0483	-0.3684	-0.3262	-0.0973	0.5528	-0.7239	0.1949	0.2406	0.4083	0.5602	-0.222	-0.0531	0.0394
-0.5352	-1.2287	-0.812	-0.6193	0.3615	-1.8907	0.0423	0.1488	0.737	0.8379	-0.5736	-0.5099	-0.2691
-0.0619	-0.0274	-0.2956	-0.054	0.4121	-0.1804	-0.0739	-0.0436	-0.101	0.0155	-0.2136	-0.1168	-0.0694
-0.0307	-0.3523	1.2691	-0.0534	-1.0729	-0.1393	0.3118	0.3241	1.369	0.4956	1.3289	0.2153	0.2455
-0.5464	-0.0622	-0.513	-0.6138	-0.3951	-1.1001	-0.2344	-0.2089	0.1222	0.1071	-0.4753	-0.4819	-0.4225
-0.3822	-0.9477	-0.543	-0.4616	0.2113	-1.4253	0.0692	0.1384	0.5906	0.6575	-0.3961	-0.3411	-0.1883
-0.1299	-0.4877	-0.4505	-0.1655	0.1404	-0.7864	0.1566	0.1834	0.2901	0.4995	-0.4203	-0.1291	-0.0237
-0.2003	-0.5541	-0.4329	-0.2461	0.0673	-0.83	0.0708	0.096	0.2393	0.3944	-0.4023	-0.1863	-0.1035
-0.5171	-1.2323	-0.8019	-0.6016	0.4366	-1.9241	0.0816	0.1916	0.8099	0.9086	-0.5478	-0.4885	-0.2395
-0.5048	-1.2178	-0.803	-0.589	0.4588	-1.9122	0.0914	0.2014	0.81	0.9175	-0.5493	-0.4786	-0.2294
-0.5164	-1.2327	-0.815	-0.6015	0.4463	-1.9278	0.0815	0.191	0.7994	0.9083	-0.5629	-0.4893	-0.2412
-0.521	-1.2375	-0.8084	-0.6058	0.435	-1.9304	0.0782	0.1881	0.805	0.9057	-0.5548	-0.4924	-0.2437
-0.5231	-1.2369	-0.807	-0.6073	0.4274	-1.9265	0.0742	0.1836	0.8	0.8986	-0.5541	-0.4941	-0.2463

table 2.2.2 : Independent Components for campaign no. 1\4

S. No.	Variable
1	Hm. Temperature
2	Blow oxygen
3	Tap-tap time
4	Lime adition
5	Ore
6	Bath Carbon
7	Sulpher
8	Phosphorus
9	Slag Basicity
10	FeO
11	MgO
12	CaO
13	MnO

table 2.2.3 : variables selected through ICA

11. %MgO in slag

12. %FeO in slag

13. Slag Basicity

Similarly, for all campaigns ICA is performed and variables are selected. Final list of chosen variables for ICA is given in table 2.2.3 .

CHAPTER 3

CLUSTERING TECHNIQUES

Group or Cluster analysis is a primitive technique in that no assumptions are made concerning number of groups or the group structure. Grouping is done on the basis of similarities or distances.

The basic objective of cluster analysis is to discover natural groupings of the items (or variables). In turn, we must first develop a quantitative scale on which to measure the association (similarity) between objects.

Most efforts to produce a rather simple group structure from a complex data set necessarily require a measure of "closeness" or "similarity". There is often a great deal of subjectivity involved in the choice of a similarity measure. Important consideration include the nature of variables (discrete, continuous, binary) or scales of measurements (nominal, ordinal, interval, ratio) and subject matter knowledge.

When items (units or cases) are clustered, proximity is usually indicated by some sort of distance. On the other hand, variables are usually grouped on the basis of correlation coefficients or like measure of association.

As we know, the Euclidean distance between two p-dimensional observations(items) .

$x=[x_1,x_2,.....,x_p]'$ and $y=[y_1,y_2,.....,y_p]$ is given by

$$d(x,y) = \sqrt{\left((X_1-Y_1)^2+(X_2-Y_2)^2+.....+(X_p-Y_p)^2\right)} = \sqrt{((X-Y)'(X-Y))}$$

The statistical distance between the same two observations is of the form

$$d(x,y) = \sqrt{((X-Y)'A(X-Y))}$$

Ordinarily, the entries of A^{-1} are sample variances and covariances. However, without prior knowledge of the distinct groups, these quantities cannot be computed. For this reason, Euclidean distance is often preferred for clustering.

Another distance measure is the Minkowski metric

$$d(x,y)=\left[\sum_{i=1}^{p}|x_i-y_i|^m\right]^{1/m}$$

For m=1, d (x, y) measures the "city-block" distance between two points in p dimensions. For m=2, d (x, y) becomes the Euclidean distance. In general, varying m changes the weight given to larger and smaller differences.

Whenever possible, it is advisable to use "true" distances, that is, distances satisfying the following distance properties

$d(x,y)=d(y,x)$

$d(x,y)>0$ if $x \neq y$

$d(x,y)=0$ if $x=y$

$d(x,y) \leq d(x,z)+d(z,y)$ triangle inequality

There are various clustering methods available, only a few are dealt in the thesis. Clustering is done using statistical methods as well as neural network based algorithms. In statistical methods, K-means clustering method is used, which uses Euclidean distance as similarity measure .In neural algorithms, Kohenon's Self Organising Map (som), and Adaptive Resonance Theory networks (ART) are used.

3.1 K-means clustering method

K-means method is a non -hierarchical clustering method, which is designed to group *items*, rather than *variables*, into a collection of K -clusters, the number of clusters K is specified in advance.

MacQueen suggests the term K-means for describing his algorithm that assigns each item to the cluster having the nearest centroid (mean). In its simplest version, the process is composed of these three steps.

1.Partition the items into K initial clusters.

2. Proceed through the list of items, assigning an item to the cluster whose centroid (mean) is nearest. (Distance is usually computed using Euclidean distance) Recalculate the centroid for the cluster receiving the new item and for the cluster losing the item.

3.Repeat Step2 until no more reassignments take place.

Rather than starting with a partition of all items into K preliminary groups in Step 1, we could specify k initial centroids called seed points and then proceed to Step 2.

The final assignment of the item to the cluster will be, to some, extent, dependent upon the initial partition or the initial selection of the seed points. Experience suggests that most major changes in assignment occur with the first reallocation step.

The greatest drawback of K-means clustering method is the prior determination of number of clusters K. There are strong arguments for not fixing the number of clusters, K, in advance. These include the following

1.If two or more seed points inadvertently lie within a single cluster, their resulting cluster will be poorly differentiated.

2.The existence of an outlier might produce at least one group with very disperse items.

3.Even if the population is known to consist of K groups, the sampling method may be such that data from the rarest group do not appear in the sample. Forcing the data into K groups would lead to nonsensical clusters.

In cases, where a single run of the algorithms requires the user to specify K, it is always good idea to rerun the algorithm for several choices.

3.1.1 Results of K-means clustering

For all the clustering techniques, only input features (variables) have been used for grouping of samples. However, for easy understanding of the results, actual no. of runs of converter has been used in preparation of tables.

(i) for ICA data

K-means algorithm has been applied for 4 values of k. In table 3.1.1 results for all 4 values of k is presented. For k=2, actual no. of runs for converter 563,499,595 and 937 form one group while remaining form other group. Interesting point is that both lowest and highest no. of runs belong to same group. It is the case with for all 4 values of k. but the reason behind this cannot be investigated here. For k=2, 772, 560, 539, 532, 662, 607, 712, 724, 746, 546 and 652 belong to same group, but as the value of k is increased, refinement in partition takes place. For k =3, 772, 560, 662, 724, 746, and 652 form a different cluster. Another interesting observation is that for k=3, 595 belongs

runs	k=2	k=3	k=4	k=5
772	2	3	1	1
560	2	3	1	1
563	1	1	2	2
499	1	1	2	2
937	1	1	2	2
595	1	2	2	2
539	2	2	3	3
532	2	2	3	3
662	2	3	4	4
607	2	2	1	1
712	2	2	3	3
724	2	3	4	4
746	2	3	4	5
546	2	2	3	3
652	2	3	4	4

table 3.1.1: results of k-means clustering
ica data

no. of runs	belongs to cluster			
	k=2	k=3	k=4	k=5
772	2	3	1	1
560	2	3	4	4
563	1	1	2	2
499	1	1	2	2
937	1	1	2	2
595	1	2	2	2
539	2	2	3	3
532	2	2	3	3
662	2	3	4	5
607	2	2	4	4
712	2	2	3	3
724	2	3	4	5
746	2	3	4	5
546	1	2	3	3
652	2	3	4	5

table 3.1.2: results of k-means clustering
for pca data

no .of runs	belongs to cluster			
	k=2	k=3	k=4	k=5
772	2	3	1	1
560	2	3	1	1
563	1	1	2	2
499	1	1	2	2
937	1	1	2	2
595	1	2	3	3
532	2	2	3	3
662	2	3	1	4
607	1	2	3	3
712	2	2	4	5
724	2	3	1	4
746	2	3	1	4
539	2	2	4	5
546	2	2	4	5
652	2	3	1	4

table 3.1.3 : results of k-means clustering
for r&d data

to the group containing 539, 532etc. But for k=4 and 5, it once again comes back to the group containing 499, 937 etc. For k=5, 746 form a lone cluster. For k=2 and 3, 607 was not in the 772, 560 group, but for k=4 and 5, this sample belongs to the cluster containing these samples.

(ii) for PCA data

For PCA data also k-means algorithm has been applied for 4 values of k. In table 3.1.2 the results of all 4 values of k are presented. For k=2, 563,499,937,595 and 546 belong to one group and remaining form other group. For k=3, 546 and 595 no longer belong to their old group and enter the group which contains 539,532,607 and 712. 563,499, 937 once again form same cluster. Same is the case with converter lives 772,560,662,724,746 and 652. An interesting observation here is that for k=4 and 5, 595 comes back to its old cluster i.e. cluster belonging to 499,937 and 563. For these two values of k, 772 form a lone cluster. 532,539,546 and 712 belong to other cluster. For k=4, 560,607,662,724,746 and 652 belong to one group but for k=5, 652,662,724,746 form another cluster and only 560 and 607 are left to previous one.

(iii) for rd data

The result of k-mean clustering for rd data is presented in the table 3.1.3. From this table it is revealed that though for k=2, 595 and 607 belong to the group having converter lives 563,499 and 937 but for other values of k they are not in that group. They themselves are always in one group. For k=3 and 4, 772, 560, 662, 724, 746 and 652 belong to same group but for k=5, 772 and 560 form other cluster. For k=4 and 5, 712, 539,546 belong to same group.

3.2 Fuzzy C-means Clustering

Bezdek developed an extremely powerful classification method to accommodate fuzzy data. It is an extension of method known as C-means (or K-means) clustering, when employed in a crisp classification sense. To introduce this method, we define a sample set of n data samples that we wish to classify:

$$X=\{x_1,x_2,\ldots\ldots\ldots\ldots,x_n\}$$

Each data sample, x_i is defined by m features, i.e.

$$x_i = \{x_{i1},x_{i2},x_{i3},\ldots\ldots\ldots\ldots,x_{im}\}$$

where each x_i in the universe X is an m-dimensional vector of m elements or m features. Since the m features can have different units, in general, we have to normalize each of features to a unified scale before classification. In a geometric sense, each $\mathbf{x_i}$ is a point in m-dimensional feature space, and the universe of the data sample, X, is a point set with n elements in the sample space.

In Fuzzy C-mean (FCM) clustering method, we define a family of fuzzy sets { A_i , i = 1,2,.....,c }as a fuzzy c-partition on a universe of data points, X. Because fuzzy sets allow for degrees of membership we can extend the crisp classification idea into a fuzzy classification notion. Then we can assign membership in more than one class. It will be useful to describe the membership value that the k^{th} data point has in the i^{th} class with the following notation:

$$\mu_{ik} = \mu_{A_i}(x_k) \in [0,1]$$

with the restriction (as with crisp classification) that the sum of all membership values for a single data point in all of the classes has to be unity:

$$\sum_{i=1}^{c} \mu_{ik} = 1 \qquad \text{for all } k = 1,2,.....,n$$

As in crisp classification, there can be empty classes and there can be no class that contains all data points. This qualification is manifested in the following expression:

$$0 < \sum_{k=1}^{n} \mu_{ik} < n$$

Because each data point can have partial membership in more than one class, i.e.

$$\mu_{ik} \wedge \mu_{jk} \neq 0$$

and

$$\bigvee_{i=1}^{c} \mu_{A_i}(x_k) = 1 \qquad \text{for all } k$$

$$0 < \sum_{k=1}^{n} \mu_{A_i}(x_k) < n \qquad \text{for all I}$$

We now define a family of fuzzy partition matrices, M_{fc} , for the classification involving c classes and n data points,

$$M_{fc} = \{ U \mid \mu_{ik} \in [0,1]; \sum_{i=1}^{c} \mu_{ik} = 1; 0 < \sum_{k=1}^{n} \mu_{ik} < n \}$$

where i = 1,2,3,.....c and k = 1,2,3,.....n .

Any $U \in M_{fc}$ is a fuzzy partition, and it follows from the overlapping character of the classes and infinite number of membership values possible for describing class membership that the cardinality of M_{fc} is also infinity.

To describe a method to determine the fuzzy c-partition matrix U for grouping a collection of n data sets into c classes, define an objective function J_m for a fuzzy c-partition,

$$J_m(U,v) = \sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{ik})^{m'}(d_{ik})^2$$

where

$$d_{ik} = d(x_k - v_j) = \left[\sum_{j=1}^{m}(x_{kj} - v_{ij})^2\right]^{1/2}$$

and where μ_{ik} is the membership of k^{th} data point in i^{th} class.

The function J_m can have a large number of values, the smallest one associated with the best clustering. Because of large number of possible values, now infinite because of the cardinality of fuzzy sets, we seek to find the best possible, or optimum, solution without resorting to an exhaustive, or expensive, search. The distance measure d_{ik} is a Euclidean distance between the i^{th} cluster center and the k^{th} data set (data point in m-space). The parameter m', is called weighting parameter, which has a range $m' \in [1,\infty)$. This parameter controls the amount of fuzziness in classification process. The vector v_i is the i^{th} cluster center, which is described by m features (m coordinates) and can arranged in vector form as, $v_i = \{v_{i1}, v_{i2}, v_{i3}, \ldots v_{im}\}$.

Each of the cluster coordinates for each class can be as follows

$$v_{ij} = \frac{\sum_{k=1}^{n}\mu_{ik}^{m'}.x_{kj}}{\sum_{k=1}^{n}\mu_{ik}^{m'}}$$

where j is a variable on feature space i.e., j =1,2,.....m.

Optimum fuzzy c-partition will be the smallest of the partitions described by function J_m , i.e.

no .of runs	belongs to cluster with membership			
	1	2	3	4
772	0.1248	0.8175	0.0378	0.0199
560	0.0243	0.957	0.0133	0.0054
563	0.1251	0.0842	0.0986	0.6921
499	0.085	0.0682	0.1314	0.7153
937	0.0948	0.0542	0.1216	0.7294
595	0.1144	0.0857	0.6129	0.187
532	0.0573	0.066	0.8379	0.0388
662	0.1109	0.0567	0.7819	0.0506
607	0.8474	0.0646	0.0492	0.0387
712	0.0932	0.7554	0.1109	0.0405
724	0.1793	0.0695	0.6995	0.0518
746	0.8697	0 0622	0 0495	0.0187
539	0.6317	0.1976	0.0973	0.0733
546	0.031	0.0309	0.9162	0.0219
652	0.7595	0.1318	0.0817	0.0269

table 3.2.1 : results of fuzzy C-means clustering for ica data (c=4)

no .of runs	belongs to cluster with membership			
	1	2	3	4
772	0.2527	0.6051	0.0907	0.0516
560	0.0478	0 9034	0.0378	0.0109
563	0.114	0.0928	0.1307	0.6624
499	0.0693	0.079	0.1319	0.7198
937	0.0414	0.0377	0.0538	0.8671
595	0.1338	0.2234	0.3266	0.3161
532	0.0703	0.1026	0.7771	0.05
662	0.2114	0.1576	0.5399	0.0911
607	0.849	0.0644	0.0548	0.0318
712	0.0882	0.6912	0.158	0.0626
724	0.1854	0.1096	0.6466	0.0584
746	0.9332	0.0361	0.021	0.0097
539	0.7468	0.1303	0.0799	0.0431
546	0.0453	0.0797	0.8208	0.0542
652	0.6269	0.2428	0.093	0.0373

table 3.2.2 : results of fuzzy C-means clustering for pca data (c=4)

no .of runs	belongs to cluster with membership			
	1	2	3	4
772	0.055	0.0778	0.4429	0.4244
560	0.0148	0.0285	0.8559	0.1008
563	0.4813	0.2971	0.1272	0.0943
499	0.8427	0.0692	0.0546	0.0335
937	0.887	0.0534	0.0338	0.0258
595	0.1655	0.4262	0.2712	0.1371
532	0.0904	0.6403	0.1259	0.1434
662	0.0139	0.0354	0.0481	0.9025
607	0.078	0.0886	0.7511	0.0823
712	0.0435	0.8124	0.0588	0.0853
724	0.0019	0.005	0.0108	0.9824
746	0.0126	0.0333	0.054	0.9001
539	0.0298	0.9087	0.0296	0.0318
546	0.0564	0.848	0.0516	0.044
652	0.0196	0.0482	0.153	0.7792

table 3.2.3 : results of fuzzy C-means clustering for rd data (c=4)

$$J_m^*(U^*, v^*) = \min_{M_{fc}} J(U, v)$$

Here we seek the best solution available within a pre-specified level of accuracy. The following algorithm was proposed by Bezdek for this purpose.

Algorithm

[i] Fix c ($2 \le c < n$) and select a value for parameter m'. Initialize the matrix $U^{(0)}$. Each step in this algorithm will be labeled r, where r = 0,1,2,.....

[ii] Calculate the c centers $\{v_i^{(r)}\}$ for each step.

[iii] Update the partition matrix for the r^{th} step, $U^{(r)}$ as follows :

$$\mu_{ik}^{(r+1)} = \left[\sum_{j=1}^{c} \left(\frac{d_{ik}^{(r)}}{d_{jk}^{(r)}} \right)^{2/(m'-1)} \right]^{-1} \quad \text{for } I_k = \phi$$

or

$$\mu_{ik}^{(r+1)} = 0 \qquad \text{for all classes i where } i \in \bar{I}_k$$

where

$$I_k = \{ i \mid 2 \le c < n ; d_{ik}^{(r)} = 0 \}$$

and

$$\bar{I}_k = \{1,2,.....,c\} - I_k$$

and

$$\sum_{i \in \bar{I}_k} \mu_{ik}^{(r+1)} = 1$$

[iv] If $\left\| U^{(r+1)} - U^{(r)} \right\| \le \varepsilon_1$, stop; otherwise set r = r+1 and return step 2.

In step [iv] we compare the matrix norm || || of two successive fuzzy partitions to a prescribed level of accuracy, ε_1 to determine whether the solution is good enough.

3.2.1 Results of Fuzzy C-means clustering

Fuzzy C- means clustering (FCM) has also been applied for 4 values of c, namely 2,3,4 and 5. In tables 3.2.1 to 3.2.6 the membership value of the data samples for c=4 and 5 is given and tables 3.2.7 to 3.2.9 contain only cluster number to which sample belongs with maximum membership.

(i) for ICA data

no. of runs	belongs to cluster with membership				
	1	2	3	4	5
772	0.0424	0.6631	0.1394	0.0234	0.1317
560	0.0106	0.9484	0.0128	0.0044	0.0239
563	0.0917	0.0809	0.0976	0.608	0.1219
499	0.1167	0.0618	0.0593	0.6771	0.0851
937	0.0967	0.0461	0.0538	0.7168	0.0866
595	0.5271	0.0813	0.0618	0.1871	0.1427
532	0.8421	0.0486	0.0216	0.0282	0.0595
662	0.6155	0.0644	0.0462	0.0604	0.2135
607	0.0566	0.075	0.4328	0.0496	0.3859
712	0.0796	0.7582	0.0376	0.03	0.0946
724	0.5596	0.0674	0.0702	0.053	0.2499
746	0.012	0.0154	0.0211	0.0051	0.9464
539	0.0046	0.0087	0.967	0.0037	0.016
546	0.9281	0.0195	0.0108	0.014	0.0275
652	0.0112	0.0185	0.013	0.0042	0.9531

table 3.2.4 : results of fuzzy C-means clustering
for ica data (c=5)

no. of runs	belongs to cluster with membership				
	1	2	3	4	5
772	0.0445	0.7338	0.1386	0.0296	0.0536
560	0.0244	0.9141	0.0335	0.0077	0.0203
563	0.0958	0.0776	0.1012	0.5058	0.2196
499	0 1039	0.0627	0.0611	0.5807	0.1914
937	0.0227	0.0182	0 0213	0.8901	0.0477
595	0.0138	0.0103	0.0081	0.0159	0.9519
532	0.8503	0.0436	0.0341	0.0227	0.0494
662	0.2457	0.1228	0.1926	0.0739	0.365
607	0.0353	0.0634	0.8366	0.0263	0.0384
712	0.1494	0.4054	0.0954	0.0645	0.2853
724	0.4056	0.1188	0.2254	0.0663	0.1839
746	0.0155	0.0441	0.9113	0.0096	0.0195
539	0 0595	0.1548	0.686	0.0387	0.0611
546	0.8392	0 0385	0.0259	0.0294	0.067
652	0 0645	0 2536	0.559	0 0341	0 0887

table 3.2.5 : results of fuzzy C-means clustering
for pca data (c=5)

no. of runs	belongs to cluster with membership				
	1	2	3	4	5
772	0.7211	0.1775	0.0319	0.0248	0.0446
560	0.9351	0.0353	0.0093	0.0054	0.0149
563	0.0724	0.07	0.1789	0.2601	0.4185
499	0.0276	0.022	0.041	0.8493	0.0601
937	0.0276	0.0268	0.0494	0.8326	0.0637
595	0.0131	0.0116	0.0245	0.0118	0.939
532	0.08	0.1253	0.2838	0.0685	0.4424
662	0.0505	0.8919	0.0221	0.0099	0.0256
607	0.3519	0.1189	0.1096	0.1117	0.3079
712	0.0606	0.115	0.6021	0.0496	0.1727
724	0.0157	0.9743	0.0038	0.0017	0.0046
746	0.0764	0.8652	0.0252	0.0102	0.023
539	0.0081	0.0109	0.9465	0.0088	0.0258
546	0.0169	0.0188	0.8948	0.0215	0.0481
652	0.2133	0.6815	0.0381	0.0186	0.0484

table 3.2.6 : results of fuzzy C-means clustering
for rd data (c=5)

runs	c=2	c=3	c=4	c=5
772	1	1	2	2
560	1	1	2	2
563	2	3	4	4
499	2	3	4	4
937	2	3	4	4
595	2	2	3	1
539	2	2	3	1
532	2	2	3	1
662	1	1	1	3
607	1	1	2	2
712	2	2	3	1
724	1	1	1	5
746	1	1	1	3
546	2	2	3	1
652	1	1	1	5

table 3.2.7 : results of fuzzy C-means clustering for ica data (belongs with maximum membership)

no. of runs	belongs to cluster			
	c=2	c=3	c=4	c=5
772	1	3	2	2
560	1	3	2	2
563	2	2	4	4
499	2	2	4	4
937	2	2	4	4
595	2	1	3	5
539	2	1	3	1
532	1	1	3	5
662	1	3	1	3
607	1	1	2	2
712	1	1	3	1
724	1	3	1	3
746	1	3	1	3
546	2	1	3	1
652	1	3	1	3

table 3.2.8 : results of fuzzy C-means clustering for pca data (belongs with maximum membership)

no. of runs	belongs to cluster			
	c=2	c=3	c=4	c=5
772	2	2	3	1
560	2	2	3	1
563	1	3	1	5
499	1	3	1	4
937	1	3	1	4
595	1	1	2	5
532	1	1	2	5
662	2	2	4	2
607	1	2	3	1
712	1	1	2	3
724	2	2	4	2
746	2	2	4	2
539	1	1	2	3
546	1	1	2	3
652	2	2	4	2

table 3.2.9 : results of fuzzy C-means clustering for rd data (belongs with maximum membership)

The results of c-means clustering method are presented in tables 3.2.1, 3.2.4 and 3.2.7. In table 3.2.1 we have taken c=4 and the membership with which a data sample belongs to a particular group is given against corresponding converter life. For c=5, similar results are presented in table 3.2.4. In table 3.2.7, we have awarded the sample to that cluster to which it belongs with highest membership (c varies from 2 to 5). From table 3.2.1 it is clear for some data samples that they belong to other groups with a significant membership. Consider the sample for which the converter life is 539, it belong to cluster1 with the membership of 0.6317 and cluster2 with a significant value of 0.1976. But some samples clearly belong to a particular cluster with very high membership value such as 560, it belongs to cluster 2 with the membership 0f 0.957. For c=5, sample having life 595 runs belong to cluster 1 with membership 0f 0.527, cluster 4 with 0.1871 and cluster 5 with 0.1427. This represents the fuzziness in the available data.

(ii) for PCA data

The results of PCA data are presented in tables 3.2.2, 3.2.5 and 3.2.8. In tables 3.2.2 and 3.2.5 c is 4 and 5 respectively. The membership of belongingness of a pattern to a particular cluster is given against the corresponding converter life. In table 3.2.8 crisp sort of clustering is given. The pattern is awarded to a cluster to which it belongs with maximum membership. In this set we see that the sample having converter life 662 runs belongs to cluster 5 with 0.365 membership and cluster 1 with 0.2457 and it is awarded to cluster 5. Similarly, the sample having converter life of 563 runs belongs to cluster 4 with the membership 0.5058 and cluster 5 with 0.2196. Therefore we see that is really difficult to have a clear cut partition surface for this case.

(iii) for RD data

The results of FCM clustering method for RD data is presented in tables 3.2.3, 3.2.6 and 3.2.9. In table 3.2.3 the number of clusters c is kept at 4. The membership of a particular pattern for a particular cluster is given against corresponding converter life. In table 3.2.6 similar result is presented for c=5. In table 3.2.9 the pattern has been awarded to the cluster to which it belongs with maximum membership. From table

3.2.6, it is seen that sample having life 563 runs belong to cluster 5 with only 0.4185 membership value and cluster 4 with 0.2601. Similarly 532 and 607 have maximum belongingness much less than 0.5. Therefore here also we can say that a clear-cut partition surface can not be created for the available data set. In fact, all three data sets are not accurate. These are a good example of fuzzy sets.

3.3 Kohenon Self-Organizing Network

The term *self-organization* refers to the ability to learn and organize information without being given correct answers for input patterns. Thus, self-organizing networks perform unsupervised learning.

The Kohenon network consists of single layer of nodes (plus an input layer). Each node receives input from the other nodes within layer. When we build a kohenon network, it is important to properly initialize the weight vectors of the nodes. It is also advised to normalise both input vectors and weight vectors to a constant (typically unit length). Each node computes by taking the dot product of its weight vector and the input vector. The result reflects their similarity (or distance). Symbolically,

$$O_j = \mathbf{X}.\mathbf{W}_j$$

where O_j is activation level of unit j, $\mathbf{X}$ is the input vector, $\mathbf{W}_j$ is the weight vector of unit j. Suppose we place all weight vectors in a matrix called a matrix $\mathbf{W}$, and let the vector $\mathbf{O}$ represent the activations of all nodes. Then we obtain

$$\mathbf{O}=\mathbf{XW}$$

Kohenon network can perform various functions namely : clustering, learning, statistical modelling, and topology preservation. Our interest lies only in clustering provided by kohenon network, therefore we will discuss only clustering.

Clustering

Clustering is concerned with the grouping of objects (input patterns) according to their similarity. Now we discuss how kohenon network can perform clustering through a competitive learning mechanism called" winner takes all ".

In essence, the node with largest activation level is declared the winner in the competition. This node is the only node that will generate an output signal and all other

nodes are suppressed to the zero activation level. Furthermore, this node its neighbours are the only nodes permitted to learn for current input pattern.

The kohenon network uses interlayer connections to moderate connections to moderate this competition. The output of each node acts as an inhibitory input to other nodes but is actually excitatory in its neighbourhood. Thus, even though there is only one winner node, more than one node are allowed to change their weights. This complex scheme for moderating competition within a layer is known as *lateral inhibition*. The inhibitory effect of a node can also decrease also with the distance from it. The exact size of neighbourhood varies as learning goes on. It starts large and is slowly reduced, making the range of change sharper and sharper. To simulate lateral inhibition, we simply take the winner. This avoids the complexity of truly implementing this mechanism.

After training, the weight vector of each node encodes the information of a group of similar input patterns. Given an input vector, it is assigned to the node with maximum activation. Since the number of nodes is fixed, the net algorithm is similar to the k-means clustering algorithm. This kind of algorithm is more noise tolerant than algorithms which do not specify the number of clusters in advance, such as ART networks. Results, however may depend on the presentation order of input data for a small amount of training data.

Learning

The winning node and its neighbours will learn by adjusting their weight vectors according to following rule:

$$\mathbf{W}_{new} = \mathbf{W}_{old} + \eta\left(\mathbf{X} - \mathbf{W}_{old}\right)$$

where $\mathbf{X}$ is the input vector and η is the learning rate.

Since the winner's weight vector generates the largest dot product with the input vector, it means that the winning weight vector is closest to the input vector. Kohenon learning is to make the winning weight even more similar to input vector. As learning proceeds, the size of neighbourhood is gradually decreased. Fewer and fewer nodes can learn in each iteration, and finally, only the winning node learns.

no. of runs	belongs to cluster			
	2	3	4	5
772	2	3	1	1
560	2	2	1	1
563	1	1	2	2
499	1	1	2	2
937	1	1	2	2
595	1	1	3	3
539	1	2	3	3
532	1	1	3	3
662	2	3	4	5
607	2	2	2	3
712	1	1	3	4
724	1	3	4	5
746	1	3	4	5
546	2	1	2	3
652	1	2	3	4

table 3.3.1 : results of kohenon som for ica data (ns=1)

no. of runs	belongs to cluster				
	ns=0	ns=1	ns=2	ns=3	ns=4
772	1	1	1	1	1
560	1	1	1	1	1
563	2	2	3	5	5
499	2	2	3	5	5
937	2	2	4	2	1
595	2	3	5	1	3
539	3	3	5	1	3
532	3	3	3	3	5
662	5	5	2	4	5
607	4	3	4	1	1
712	3	4	1	5	4
724	4	5	2	2	1
746	5	5	1	2	1
546	3	3	5	5	5
652	4	4	2	2	1

table 3.3.2 : results of kohenon som for ica data (no. of output neurons=5)

Kohenon network uses single pass learning rather than multi pass feedback and is potentially fast. This fact suggests its suitability for real time applications.

Algorithm

- First, a winning neuron is selected as the one with shortest Euclidean distance

$$\|\mathbf{x} - \mathbf{w}_i\|$$

between its weight vector and the input vector, where $\mathbf{w}_i$ denotes the weight vector corresponding to the i^{th} output neuron.

- Let i^* denote the index of the winner and let I^* denote a set of indices corresponding to a defined neighbourhood of winner i^*. Then the weights associated with the winner and its neighbouring neurons are updated by:

$$\Delta \mathbf{w}_j = \eta \wedge (j, i^*)(\mathbf{x} - \mathbf{w}_j) \quad \text{for all } j \in I^*$$

where neighbourhood function $\wedge(j, i^*)$ may be chosen as

$$\wedge(j, i^*) = \exp\left(-\left|\mathbf{r}_j - \mathbf{r}_{i^*}\right|^2 / 2\sigma^2\right)$$

where $\mathbf{r}_j$ represents the position of the neuron j in the output space. The convergence of feature map depends on a proper choice of η. One plausible choice is that $\eta = 1/t$. The size of neighbourhood (or σ) should decrease gradually.

- The weight update should be immediately succeeded by the normalisation of $\mathbf{w}_j$.

3.3.1 Results of Kohenon SOM

Results of Kohenon SOM are presented in tables 3.3.1 to 3.3.6. Here we varied the number of output neurons (i.e. no. of clusters we want) and neighbourhood size. Neighbourhood size is playing a major role in classification. Learning rate alpha is 0.2 for all cases.

(i) for ICA data

The results for ICA data are presented in the tables 3.3.1 and 3.3.2. In table 3.3.1 neighbourhood size is kept fixed at 1. In table 3.3.2, we have kept no. of output neurons fixed at 5 and varied the neighbourhood size from 0 to 4. With reference to table 3.5.1, For 2 output neurons the samples having lives 772, 560, 662, 607 and 546 belong to one

no. of runs	belongs to cluster			
	2	3	4	5
772	2	3	1	1
560	2	2	1	1
563	1	1	2	2
499	1	1	2	2
937	1	1	1	1
595	1	1	1	4
539	2	2	3	5
532	1	1	3	3
662	1	3	4	3
607	2	1	1	4
712	1	2	2	5
724	1	3	4	2
746	1	3	4	2
546	2	1	1	4
652	1	2	3	1

table 3.3.3 : *result of kohenon som for pca data*
(ns=1)

no. of runs	belongs to cluster				
	ns=0	ns=1	ns=2	ns=3	ns=4
772	1	1	1	1	1
560	1	1	1	3	4
563	2	2	3	5	5
499	2	2	3	5	5
937	2	1	1	2	1
595	2	4	4	1	4
539	3	5	5	1	5
532	3	3	3	3	1
662	3	3	3	2	1
607	4	4	5	5	5
712	3	5	1	1	1
724	5	2	2	1	1
746	5	2	2	1	1
546	4	4	5	5	5
652	5	1	3	2	1

table 3.3.4 : results of kohenon som for pca data
(no. of output neurons=5)

cluster and the rest belong to another. For 3 output neurons, data samples having lives 724, 772, 746 and 662 belong to same cluster. Another cluster is formed for samples having lives 563, 499, 937, 595, 532, 712, and 546. Third cluster contains samples for which lives are 560, 539, 607 and 652. For 4 output neurons case, the winner neuron for lives 772 and 560 is 1. Neuron no. 2 wins for samples having lives 563, 499, 937, 607 and 546. Samples having lives 595, 539, 532 712 and 652 form a cluster and another cluster is formed for the samples for which lives are 662, 724 and 746. For 5 output neuron case slight modification takes place in partition. 546 now belongs to the group having lives 595, 539, 532 and 607 and 712 and 652 leave this group. All these results are given in table 3.3.1. In table 3.5.2 when we vary the neighbourhood size we see that partition line also changes even for same no. of output neurons. This may also happen that a particular neuron may not win even for single instance. Therefore it is very necessary to choose a suitable neighbourhood size for true partition.

(ii) for PCA data

The results of Kohenon SOM for PCA data are given in tables 3.3.3 and 3.3.4. In table 3.3.3 neighbourhood size is kept fixed at 1 and no .of output neurons is varied. In table 3.3.4 neighbourhood size is varied from 0 to 4 for 5 output neurons. As observed from table 3.3.3, for 2 output neurons (i.e. when we want 2 clusters) the patterns for which converter lives are 772, 560, 539, 607 and 546 runs form one group and rest form another. When no. of output neurons is increased to 3, modification in partition surface takes place. Patterns having converter lives 563, 499, 937, 595, 532, 607 and 546 runs belong to one group. Another group contains lives 772, 662, 724 and 746. Rest of the patterns form third group. This classification may vary if we change the neighbourhood size. As is seen from table 3.3.4 , when neighbourhood size is 0 neuron the partition was such that converter lives 772, 560 were in one group. Another group 563, 499, 937 and 595 runs. As we change the initial neighbourhood size to 1 neuron, 937 entered the group having lives 772 and 560. Similarly for neighbourhood size of 2 neurons 712 also enters this group while 652 leaves it.

(iii) for RD data

no. of runs	belongs to cluster	
	4	5
772	1	1
560	1	1
563	3	3
499	2	2
937	2	2
595	4	4
532	4	5
662	4	5
607	1	2
712	3	3
724	4	5
746	4	5
539	1	2
546	1	1
652	3	3

table 3.3.5 : results of kohenon som for rd data (ns=3)

no. of runs	belongs to cluster				
	ns=0	ns=1	ns=2	ns=3	ns=4
772	1	1	1	1	1
560	1	1	1	1	3
563	2	2	2	3	5
499	2	2	2	2	5
937	2	2	2	2	5
595	3	3	3	4	2
532	3	3	3	5	2
662	4	4	4	5	1
607	3	3	2	2	4
712	5	5	5	3	1
724	4	4	4	5	1
746	4	4	4	5	1
539	5	5	5	2	3
546	5	5	5	1	4
652	4	4	3	3	1

table 3.3.6 : results of kohenon som for rd data (no. of output neurons=5)

The results for RD data are presented in tables 3.3.5 and 3.3.6. In table 3.3.5 neighbourhood size is kept fixed at 3 and no. of output neurons are varied. In one case it is 4 while in other it is 5 neurons. In table 3.3.6 we have varied neighbourhood size keeping number of output neurons fixed at 5. The clusters are almost similar in both the cases in table 3.3.5 except two differences. 595 makes a separate cluster in the case with 5 neurons. Also 539 and 607 leave their group and are merged in the group containing samples having lives 499 and 937 runs. As we vary the neighbourhood size (from 0 neuron to 4 neurons) in the case of 5 output neurons, we observe that first three case give same result. Changes occur only in last two cases. So we can say here that a good partition takes place when neighbourhood size is much smaller than number of output neurons.

3.4 ART2 (Continuos valued ART)

ART2 is widely used clustering technique for analog or continuous valued patterns. The patterns are classified or clustered with the accuracy defined by a factor called "vigilance factor". Smaller the vigilance threshold larger the number of clusters generated Their ability to generalise is limited, however, the ability of this network to create new pattern class in its knowledge base on the arrival of a novel pattern makes it very suitable for clustering (i.e. these networks are unsupervised). The classification is dependent on the order of presentation of input patterns.

Let $\mathbf{x}$ and $\mathbf{w}_j$ denote the input vector and the weight of neuron j respectively. The criterion of selecting the winner is based on minimum distance measure (e.g. Euclidean or other distance).

Algorithm

- Given a new training pattern, a winner neuron is selected with minimum distance criterion. Suppose the winner neuron is j^*, then

$$\|\mathbf{x} - \mathbf{w}_{j^*}\| \leq \|\mathbf{x} - \mathbf{w}_j\| \qquad \text{for all } j$$

where $\|\ .\ \|$ denotes the distance metric (Euclidean or any other distance).

- *Vigilance test*: A neuron j^* passes the vigilance test if and only if

$$\|\mathbf{x} - \mathbf{w}_{j^*}\| < \rho$$

no. of runs	belongs to cluster																									
	2.5	2.6	2.7	2.8	2.9	3	3.1	3.2	3.3	3.4	3 5	3.6	3.7	3 8	3.9	4	4.1	4.2	4 3	4.4	4.5	4.6	4.7	4.8	4.9	5
772	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
560	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
563	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
499	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2
937	4	4	4	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2
595	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	2	2	2	2	2	2	2	2
539	6	6	6	6	6	6	6	6	6	6	5	5	4	4	4	4	4	4	4	4	4	4	3	3	2	2
532	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	3	3	2	2
662	7	7	7	7	7	7	7	7	7	7	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
607	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
712	6	6	6	6	6	6	6	6	6	6	5	5	4	4	4	4	4	4	4	4	4	4	3	3	2	2
724	7	7	7	7	7	7	7	7	7	7	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
746	7	7	7	7	7	7	7	7	7	7	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
546	6	6	6	6	6	6	6	6	6	6	5	5	4	4	4	4	4	4	4	4	4	4	3	3	2	2
652	7	7	7	7	7	7	7	7	7	7	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

table 3.4.1 : results of ART2 network for ica data
(vigilance parameter is varied from 2.5 to 5)

no. of runs	belongs to cluster																									
	2.5	2.6	2.7	2.8	2.9	3	3 1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3 9	4	4.1	4.2	4.3	4 4	4.5	4 6	4 7	4.8	4.9	5
772	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
560	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
563	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
499	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
937	4	4	4	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
595	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3
539	6	6	6	6	6	6	6	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4
532	7	7	7	7	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
662	8	8	8	8	7	7	7	6	6	6	6	6	6	6	6	1	1	1	1	1	1	1	1	1	1	1
607	9	9	9	9	8	8	8	1	1	1	1	1	1	1	1	4	4	1	1	1	1	1	1	1	1	1
712	7	7	7	7	5	5	5	4	4	4	4	4	4	4	4	5	5	4	4	4	4	4	4	4	4	4
724	8	8	8	8	7	7	7	6	6	6	6	6	6	6	6	1	1	1	1	1	1	1	1	1	1	1
746	8	8	8	8	7	7	7	6	6	6	6	6	6	6	6	1	1	1	1	1	1	1	1	1	1	1
546	6	6	6	6	6	6	6	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4
652	8	8	8	8	7	7	7	6	6	6	6	6	6	6	6	1	1	1	1	1	1	1	1	1	1	1

table 3.4.2 : results of ART2 network for pca data
(vigilance parameter is varied from 2.5 to 5)

where the vigilance value ρ determines the radius of cluster.

- If the winner fails the vigilance test, a new neuron unit k is created with weight $w_k = x$.
- If the winner passes the vigilance test, adjust the weight of winner j^* by

$$w_{j^*}^{new} = \frac{x + w_{j^*}^{old} \left\| cluster_{j^*}^{old} \right\|}{\left\| cluster_{j^*}^{old} \right\| + 1}$$

where $\left\| cluster_i \right\|$ denotes the number of members in $cluster_i$.

3.4.1 Results of ART2

For all three data sets, vigilance factor is varied between 2.5 to 5.0 . For all three data sets, all the variables are scaled between 0 and 1. In table 3.4.1, result for ICA data is given. In tables 3.4.2 and 3.4.3 results for PCA data and RD data are presented. The number of cluster reduces as vigilance factor is increased.

(i) for ICA data

For vigilance factor 2.5, there are 7 clusters present. For vigilance up to 3.4, there is no change in clustering. For vigilance 3.5, there are 6 clusters. The no. of cluster further reduces to 5 and 4 for vigilance factors 3.6 and 3.7 respectively. Then up to vigilance 4.6 there are 4 clusters. For 4.7 and 4.8 there are 3 clusters and for vigilance factors 4.9 and 5.0 there are only two clusters available. For all the values of vigilance factors, samples having lives 772,560 and 662 belong to same group. For a small value of vigilance factor, there are many groups that contain only one member. In contrast to all previous methods discussed, in ART2 clustering 499 and 937 belong to different group. According to this method, 499 belong to altogether a different cluster. Even for vigilance as high as 4.6 it forms a separate group. For vigilance 4.7 and above 563,499,937 and 595 belong to same cluster as happened with both K-means and Fuzzy C-means clustering. For all values of vigilance factor, 652,662, 724 and 746 belong to same group. Similarly, 539, 546 and 712 are always in same group.

(ii) for PCA data

For vigilance factor 2.5, there are 9 clusters in this case. The classification remains same till vigilance is 2.8. Here 772 and 560 belong to one cluster. 563, 499, 937, 595, 539, 607 are the only members of their respective clusters. 532 and 712 form

belongs to cluster																									
2.5	2.6	2.7	2.8	2.9	3	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	4	4.1	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4 9	5
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2
5	5	5	4	4	4	4	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2
6	6	6	5	5	5	5	4	4	4	4	4	4	2	2	2	2	3	3	3	3	2	2	2	2	2
7	7	7	6	5	5	5	4	4	4	4	4	4	2	2	2	2	3	3	3	3	2	2	2	2	2
8	8	8	7	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	9	9	8	7	6	6	5	5	5	4	4	4	4	2	2	2	3	3	3	3	2	2	2	2	2
7	7	7	6	5	5	5	4	4	4	4	4	4	2	2	2	2	3	3	3	3	2	2	2	2	2
8	8	8	7	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	8	8	7	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	10	7	6	5	5	5	4	4	4	4	4	4	2	2	2	2	3	3	3	3	2	2	2	2	2
10	10	7	6	5	5	5	4	4	4	4	4	4	2	2	2	2	3	3	3	3	2	2	2	2	2
8	8	8	7	6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

table 3.4.3: results of ART2 nework for rd data
(vigilance parameter is varied from 2.5 to 5)

one cluster and similarly, 662, 724, 746 and 652 belong to another cluster. As vigilance is varied to 2.9, modification in classification takes place and there are now 8 clusters. This is the case till vigilance is 3.1. 532 and 595, which were forming separate clusters previously, are now merged together. Rest of the classification is similar to the previous case. For vigilance 3.2 to 3.9, there are 6 clusters. There must have been a classification boundary between 3.1 and 3.2 that we did not notice. Here, 607, which was forming a separate cluster previously, is now merged in the group containing 772 and 560. 563 and 937 are merged together. 712 is now in the 539, 595 group. For vigilance 4 and 4.1 there are 5 clusters. In cluster no. 1, there are samples for which converter lives are 772,560, 662, 712, 724 and 652. Cluster no. 2 contains data samples having lives 563 and 937. In cluster no. 3 only 499 is present. In cluster no. 4 there are samples having lives 595 and 532. Samples having lives 712, 539 and 546 belong to cluster no.5.For vigilance from 4.2 to 5.0 there are 4 clusters. No. of clusters could have been further reduced if we had increased the vigilance factor.

(iii) for RD data

Table 3.4.3 contains the results of ART2 applied for RD data for vigilance factors 2.5 to 5.0 with a step of 0.1. The number of cluster varies from 10 to 2 for given values of vigilance factors. For vigilance factors 2.5 and 2.6, there are 10 clusters. All the data samples form separate cluster except the samples having lives 652, 662. 724, 746 that form a cluster and the samples having lives 539, 546, 532 and 712.539 and 546 belong to one cluster while 532 and 712 belong to another. For vigilance factor 2.7 there are 9 clusters. The two groups which contained converter lives 539,546 and 532,712 are now merged. Rest of the classification remains the same. For vigilance factor 2.8, there are 8 clusters. Samples having lives 772 and 560 now belong to same cluster. For vigilance 2.9, data set is divided into 7 clusters. For vigilance 3 and 3.1 there are 6 clusters whereas for 3.2 to 3.4 data set contains 5 clusters. For vigilance factors 3.5 to 3.8 there are 4 groups. Though for vigilance 3.8, no. of cluster remains 4, but data partition is different as compared to others. This is only because of vigilance factor. For vigilance 3.9 to 4.5, data set is partitioned in to 3 clusters. For remaining vigilance factors there are only two clusters.

3.5 FUZZY ART

Adaptive Resonance Theory (ART) networks are most useful for pattern clustering, classification, and recognition. Their ability to generalise is limited, however, the ability of these networks to create a new pattern class in its knowledge base on the arrival of novel pattern makes them very useful. These networks can work on binary or analog value patterns.

Fuzzy ART is a form of ART1 that incorporates fuzzy logic operations. Although ART1 can learn to classify only binary input patterns, fuzzy ART can learn to classify both analog and binary input patterns.

In fuzzy ART, input vectors are normalised at a preprocessing stage. This normalisation procedure, called complement coding, leads to a symmetric theory in which the MIN operator ($\wedge$) and the MAX operator ($\vee$) of fuzzy set theory play complementary roles.

Algorithm

ART field activity vectors - Fuzzy ART system includes a field F_0 of nodes that represent a current input vector; a field F_1 that receives both bottom-up input from F_0 and top-down input from field F_2 that represents the active code, or category. The F_0 activity vector is denoted $I = (I_1, I_2, \ldots\ldots, I_M)$ with each component I_i in the interval $[0,1]$, $i = 1,2,\ldots\ldots M$. The F_1 activity vector is denoted $\mathbf{x} = (x_1, x_2, \ldots\ldots, x_M)$ and the F_2 activity vector is denoted $y = (y_1, y_2, \ldots\ldots, y_N)$.

Weight Vector - Associated with each F_2 category node j ($j = 1,2,\ldots\ldots N$) is a vector $\mathbf{w}_j \equiv (w_{j1}, \ldots\ldots, w_{jM})$. Initially,

$w_{j1}(0) = w_{j2}(0) = \ldots\ldots = w_{jM}(0) = 1$.

Parameters - Fuzzy ART dynamics are determined by a choice parameter $\alpha > 0$; a learning rate parameter $\beta \in [0,1]$; and a vigilance parameter $\rho \in [0,1]$.

Category choice - For each input $\mathbf{I}$ and F_2 node j, the choice function T_j is defined by:

$$T_j(\mathbf{I}) = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|},$$

where the fuzzy AND operator $\wedge$ is defined by :

$$(p \wedge q)_i \equiv \min(p_i, q_i)$$

and where the norm |.| is defined by :

$$|p| \equiv \sum_{i=1}^{M} |p_i|$$

for any M-dimensional vectors **p** and **q**. For notational simplicity, $T_j(I)$ is often written as T_j when the input **I** is fixed.

The system is said to make a category choice when at most one F_2 node can become active at a given time. The category choice is indexed by J, where $T_j = \max\{ T_j : j = 1,2,.....N\}$.

If more than one T_j is maximal, the category j with the smallest index is chosen. In particular, nodes become committed in order j = 1,2,3,..... When the J^{th} category is chosen, $y_J = 1$; and $y_j = 0$ for $j \neq J$. In a choice system, the F_1 activity vector x obeys the equation

$$x = \begin{cases} \mathbf{I} & \text{if } F_2 \text{ is inactive} \\ \mathbf{I} \wedge \mathbf{w}_j & \text{if the } j^{th} \text{ node is chosen} \end{cases}$$

Resonance or reset - Resonance occurs if the match function $|\mathbf{I} \wedge \mathbf{w}_j| / |\mathbf{I}|$ of the chosen category meets the vigilance criterion :

$$\frac{|\mathbf{I} \wedge \mathbf{w}_J|}{|\mathbf{I}|} \geq \rho$$

i.e. when the j^{th} category is chosen, resonance occur if

$$|x| = |\mathbf{I} \wedge \mathbf{w}_J| \geq \rho |\mathbf{I}|$$

Mismatch occur if

$$\frac{|\mathbf{I} \wedge \mathbf{w}_J|}{|\mathbf{I}|} < \rho$$

i.e. if

$$|x| = |\mathbf{I} \wedge \mathbf{w}_J| < \rho |\mathbf{I}|$$

Then the value of the choice function T_J is set to 0 for the duration of the input presentation to prevent the persistent selection of same category during search. A new

no. of runs	belongs to cluster									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
772	1	1	1	1	1	1	1	1	1	1
560	1	1	1	1	1	1	1	1	1	2
563	1	1	1	1	2	2	2	2	2	3
499	1	1	1	1	2	2	2	3	3	4
937	1	1	1	2	2	2	3	4	4	5
595	1	1	1	2	2	3	4	5	5	6
539	1	1	1	2	1	3	4	6	6	7
532	1	1	1	2	1	3	4	5	7	8
662	1	1	1	1	1	1	1	7	8	9
607	1	1	1	1	1	1	1	8	9	10
712	1	1	1	2	1	3	5	9	10	11
724	1	1	1	1	1	1	6	10	11	12
746	1	1	2	1	1	1	6	11	12	13
546	1	1	2	2	3	3	4	6	13	14
652	2	2	3	3	4	4	7	12	14	15

table 3.5.1 : results of fuzzy ART network for ica data
alpha=0.6,beta=0.6, vig is varied

no. of runs	belongs to cluster									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
772	1	1	1	1	1	1	1	1	1	1
560	1	1	1	1	1	1	1	1	1	1
563	2	2	2	2	2	2	2	2	2	2
499	2	2	2	2	2	2	2	2	2	2
937	2	2	2	2	2	2	2	2	2	2
595	3	3	3	3	3	3	3	3	3	3
539	3	3	3	3	3	3	3	3	3	3
532	3	3	3	3	3	3	3	3	3	3
662	1	1	1	1	1	1	1	1	1	1
607	1	1	1	1	1	1	1	1	1	1
712	3	3	3	3	3	3	3	3	3	3
724	1	1	1	1	1	1	1	1	1	1
746	1	1	1	1	1	1	1	1	1	1
546	3	3	3	3	3	3	3	3	3	3
652	4	4	4	4	4	4	4	4	4	4

table 3.5.2 : results of fuzzy ART network for ica data
vig=0.5,beta=0.6, alpha is varied

index J is then chosen. The search continues until the chosen J satisfies resonance condition.

Learning - Once search ends, the weight vector $\mathbf{w}_j$ is updated according to the equation

$$\mathbf{w}_J^{new} = \beta\left(\mathbf{I} \wedge \mathbf{w}_J^{old}\right) + \left(1-\beta\right)\mathbf{w}_J^{old}$$

Input normalisation / complement coding option - Proliferation of categories is avoided in fuzzy ART if inputs are normalised rule that preserved amplitude information. To define this operation in its simplest form, let **a** represent on response, then complement of **a**, denoted by $\mathbf{a}^c$, represents the off-response, where

$$a_i^c \equiv 1 - a_i$$

The complement-coded input **I** to the field F_1 is the 2M- dimensional vector :

$$\mathbf{I} = (a, a^c) \equiv \left(a_1, \ldots\ldots\ldots\ldots, a_M, a_1^c, \ldots\ldots\ldots\ldots, a_M^c\right)$$

Note that

$$|\mathbf{I}| = \left|\left(\mathbf{a}, \mathbf{a}^c\right)\right|$$

$$= \sum_{i=1}^{M} \mathbf{a}_i + \left(M - \sum_{i=1}^{M} \mathbf{a}_i\right)$$

$$= M$$

so inputs are preprocessed into complement coding form are automatically normalised. Where complement coding is used, the initial weight vectors are given by

$$\mathbf{w}_{j1}(0) = \ldots\ldots\ldots\ldots = \mathbf{w}_{j,2M} = 1$$

3.5.1 Results of Fuzzy ART network

The results of Fuzzy ART network are presented in tables 3.5.1 to 3.5.6. The vigilance factor and alpha are varied over range [0.1,1]. For all three data sets alpha is having no effect on clustering. As expected, vigilance factor plays major role. Larger the vigilance factor, larger is the number of clusters created.

(i) for ICA data

Results of clustering obtained by Fuzzy ART algorithm are presented in tables 3.5.1 and 3.5.2. In table 3.5.1 we have varied vigilance factor from 0.1 to 1 and in table 3.5.2 alpha is varied over the same range. For vigilance value 0.1 and 0.2 we have two clusters. All the samples except the one which is having converter life 652 belong to

no. of runs	belongs to cluster									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
772	1	1	1	1	1	1	1	1	1	1
560	1	1	1	1	1	1	1	1	2	2
563	1	1	1	1	2	2	2	2	3	3
499	1	1	1	2	2	2	2	3	4	4
937	1	1	1	2	2	2	2	2	5	5
595	1	1	1	2	2	2	3	4	6	6
539	1	1	2	2	1	3	4	5	7	7
532	1	1	2	1	1	3	3	6	8	8
662	1	1	2	1	1	1	5	7	9	9
607	1	1	2	1	3	3	3	8	10	10
712	1	1	2	2	3	3	4	6	11	11
724	1	1	2	1	1	1	5	9	12	12
746	1	1	2	1	1	4	5	9	13	13
546	2	2	3	3	4	5	6	10	14	14
652	3	3	4	4	5	6	7	11	15	15

table 3.5.3 : results of fuzzy ART network for pca data
alpha=0.5,beta=0.6, vig is varied

no. of runs	belongs to cluster									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
772	1	1	1	1	1	1	1	1	1	1
560	1	1	1	1	1	1	1	1	1	1
563	2	2	2	2	2	2	2	2	2	2
499	2	2	2	2	2	2	2	2	2	2
937	2	2	2	2	2	2	2	2	2	2
595	2	2	2	2	2	2	2	2	2	2
539	3	3	3	3	3	3	3	3	3	3
532	3	3	3	3	3	3	3	3	3	3
662	1	1	1	1	1	1	1	1	1	1
607	3	3	3	3	3	3	3	3	3	3
712	3	3	3	3	3	3	3	3	3	3
724	1	1	1	1	1	1	1	1	1	1
746	4	4	4	4	4	4	4	4	4	4
546	5	5	5	5	5	5	5	5	5	5
652	6	6	6	6	6	6	6	6	6	6

table 3.5.4 : results of fuzzy ART network for pca data
vig=0.6,beta=0.6, alpha is varied

no. of runs	belongs to cluster									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
772	1	1	1	1	1	1	1	1	1	1
560	1	1	1	1	1	2	2	2	2	2
563	1	1	2	2	2	3	3	3	3	3
499	1	1	2	2	2	3	3	4	4	4
937	1	1	2	2	2	3	3	4	5	5
595	1	1	2	2	2	3	4	5	6	6
539	1	1	2	2	2	4	4	6	7	7
532	1	1	1	1	1	4	5	7	8	8
662	1	2	2	2	2	2	4	8	9	9
607	1	2	2	2	2	4	6	6	10	10
712	1	2	1	1	1	4	5	9	11	11
724	1	2	1	1	3	4	5	9	12	12
746	1	2	2	2	4	5	6	6	13	13
546	1	2	2	2	4	5	6	10	14	14
652	2	3	3	3	5	6	7	11	15	15

table 3.5.5: results of fuzzy ART for rd data
alpha=0.5, beta=0.6, vigilance is varied

no. of runs	belongs to cluster									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
772	1	1	1	1	1	1	1	1	1	1
560	1	1	1	1	1	1	1	1	1	1
563	2	2	2	2	2	2	2	2	2	2
499	2	2	2	2	2	2	2	2	2	2
937	2	2	2	2	2	2	2	2	2	2
595	2	2	2	2	2	2	2	2	2	2
539	2	2	2	2	2	2	2	2	2	2
532	1	1	1	1	1	1	1	1	1	1
662	2	2	2	2	2	2	2	2	2	2
607	2	2	2	2	2	2	2	2	2	2
712	1	1	1	1	1	1	1	1	1	1
724	3	3	3	3	3	3	3	3	3	3
746	4	4	4	4	4	4	4	4	4	4
546	4	4	4	4	4	4	4	4	4	4
652	5	5	5	5	5	5	5	5	5	5

table 3.5.6: results of fuzzy ART for rd data
vig=0.6, beta=0.6, alpha is varied

single cluster. Though for vigilance factors 0.3 and 0.4 we have 3 clusters but line of partition in both the cases is different. Similar effect is seen for vigilance values 0.5 and 0.6. As we move from vigilance 0.6 to 0.7, the number of clusters increases from 4 to 7. The number of clusters are 12, 14 and 15 for vigilance factors 0.8, 0.9 and 1.0 respectively. For vigilance value 1.0, in all the cases the number of clusters created will be same as no. of data samples available.

(ii) for PCA data

The results of Fuzzy ART network for PCA data are presented in the tables 3.5.3 and 3.5.4 . Initially, for vigilance value 0.1 and 0.2, set is divided in 3 clusters and then no. of clusters gradually increases with increase in vigilance factor.

(iii) for RD data

The results of Fuzzy ART network for RD data are presented in the tables 3.5.5 and 3.5.6. For vigilance factor 0.1, we have two clusters. For vigilance values 0.2, 0.3 and 0.4, data set is divided into 3 clusters. The no. of clusters available is 5 for 0.5 vigilance and 6 for 0.6. For 0.8 vigilance, we have 11 clusters and for 0.9 and 1.0, the number of clusters is 15. In most of the cases we see that in this clustering, last data sample is forming a separate cluster. This may be because of the order of presentation of input patterns. Because in this case order of presentation of patterns matters a lot. As in the case with ICA and PCA data, alpha is having no effect on clustering.

3.6 ID3 Algorithm

ID3 is a machine learning algorithm which generates decision rules from a set of training examples. Each example is represented by a list of features.

ID3 uses a tree representation for concepts. To classify a set of examples, we start at the top of the tree and answer the questions associated with the nodes in the tree until we reach a leaf node, where the classification or decision is stored.

ID3 uses information theory to select features which give the greatest information gain or decrease of entropy. Entropy is defined as $-p\log_2 p$, where p is determined from frequency of occurrence.

The ID3 algorithm is stated as follows:

- Let N be total number of learning examples, and N_i the number of examples that belong to class i , i = 1,2,3,.....C , where C = total no. of classes. The information entropy for the problem consisting of N examples is

$$\text{entropy}(I) = \sum_{i=1}^{C} -\frac{N_i}{N}\log_2\left(\frac{N_i}{N}\right)$$

- When a feature test is performed on feature A_k, k = 1,2,.....,K, all examples are divided into J subsets, where feature A_k has J values. Assume that there are $n_{kj}(i)$ examples in subset j belonging to class i and the total number of examples in subset j is $n_{kj} = \sum_i n_{kj}(i)$

  Then the entropy of feature A_k is calculated as

$$\text{entropy}(I, A_k) = \sum_{j=1}^{J} \frac{n_{kj}}{N}\text{entropy}(I, A_k, j)$$

$$= -\sum_{j=1}^{J}\sum_{i=1}^{C} \frac{n_{kj}}{N}\left[\frac{n_{kj}(i)}{n_{kj}}\log_2\left(\frac{n_{kj}(i)}{n_{kj}}\right)\right]$$

- The new level of decision tree is built by adding the leaf nodes resulting from testing feature A_k^*, where the test on feature A_k^* results in the maximum information gain :

$$\max_{A_k^*}\left\{\text{entropy}(I) - \text{entropy}(I, A_k)\right\}$$

- Testing different features and growing a decision tree should continue until all leaf node contain examples of single class only. The corresponding entropy is then

$$\text{entropy}(I, A_k^1, A_k^2, \ldots\ldots A_k^L) = 0$$

where the decision tree consists of L ($L \le K$) levels and corresponding feature test at level l is A_k^l.

Finally, we obtain a decision tree, which can be described in terms of the hierarchical decision rules. The examples belonging same leaf node belong to same cluster.

3.6.1 Results of ID3 Algorithm

these are sorted features

0.444	2
0.647	13
0.667	9
0.8	10
0.863	12
0.876	3
0.883	4
0.907	6
0.909	11
0.918	8
0.918	5
0.943	7
0.967	1

this is classification tree

3	4	7	10	15	17	20	24	27	30	33	34	38	70	1
2	4	7	10	14	18	20	24	25	28	32	36	38	70	9
2	4	7	10	14	17	19	23	27	28	33	35	38	70	13
3	4	7	10	15	17	20	24	26	30	33	34	37	50	2
2	4	7	10	14	18	20	23	25	29	33	35	37	70	12
2	4	7	10	14	18	20	24	25	29	33	34	37	70	15
2	6	8	12	13	18	21	23	25	28	31	35	39	50	3
2	6	9	11	13	16	21	22	25	29	31	34	39	50	4
1	6	7	11	14	17	21	23	26	30	32	35	37	50	6
1	6	8	11	15	16	19	23	25	30	33	34	37	50	7
2	6	8	10	14	17	20	23	25	29	32	35	37	50	8
1	6	8	11	15	16	19	23	25	30	33	34	37	50	14
3	5	7	10	15	16	20	24	26	30	32	34	37	50	10
1	5	8	12	13	16	21	23	25	28	31	35	37	70	5
1	5	8	11	15	16	19	23	25	29	33	35	37	70	11

table 3.6.1 : results of ID3 algorithm for ICA data

Since the core part of this algorithm is entropy calculation, we are required to convert the input to the suitable form. Therefore each variable in every data set is divided into three ranges depending upon the variation in them. The converter life is divided into two classes (i) when the life is below 650 runs (ii) when life is more than 650 runs. The input to the ID3 algorithm is given in table A5 for ICA data, in table A6 for PCA data and in table A7 for RD data.

(i) for ICA data

The result of ID3 algorithm is presented in the table 3.6.1. Here classification takes three steps. Following decision rules are formulated

[1]. If mean blow oxygen is in range 6 then corresponding class of converter life is 50.

[2]. If mean blow oxygen is in range 4 and mean MnO is in range 38 then corresponding class for converter life is 70.

[3]. If mean blow oxygen is in range 4 and mean MnO is in range 37 and mean basicity is in range 26 then corresponding class of converter life is 50.

[4]. If mean blow oxygen is in range 4 and mean MnO is in range 37 and mean basicity is in range 25 then corresponding class of converter life is 70.

[5]. If mean blow oxygen is in range 5 and mean MnO is in range 37 and mean basicity is in range 26 then corresponding class for converter life is 50.

[6]. If mean blow oxygen is in range 5 and mean MnO is in range 37 and mean basicity is in range 25 then corresponding class for converter life is 70.

These ranges for variables is given in A5.

(ii) for PCA data

The result of ID3 algorithm for PCA data is presented in the table 3.6.2. Input to the algorithm is given in table A6 with corresponding ranges.

Decision takes place in 3 steps and following decision rules are formulated.

[1]. If mean Si in hot metal is in range 4 then corresponding class for converter life is 50.

[2]. If mean Si in hot metal is in range 6 then corresponding class for converter life is 50.

these are sorted features

0.4327	2
0.4437	4
0.6667	12
0.691	3
0.8	13
0.8626	14
0.8763	6
0.883	7
0.9073	9
0.9183	11
0.9183	8
0.9195	5
0.9428	10
0.9956	1

this is classification tree

2	4	9	10	13	16	19	24	26	29	33	36	39	40	70	1
1	4	9	10	14	16	19	23	27	29	33	34	37	42	70	9
1	4	8	10	14	16	19	23	27	29	32	34	38	41	70	12
1	4	9	10	14	16	19	23	26	28	32	36	37	41	70	13
1	4	8	10	13	16	19	23	27	29	33	34	38	40	70	15
1	5	8	10	14	16	19	24	26	29	33	35	39	40	50	2
2	5	7	12	14	17	21	22	27	30	32	34	37	41	50	3
3	5	7	12	13	18	20	22	25	30	31	34	38	40	50	4
1	5	7	12	13	16	20	23	26	30	32	35	39	41	50	6
1	5	8	12	14	17	19	23	26	29	32	34	38	41	50	8
1	5	7	11	13	16	19	24	25	29	33	35	39	40	50	10
3	5	8	11	13	17	21	22	25	30	32	34	37	41	70	5
1	5	8	11	14	17	20	24	25	28	32	34	38	41	70	11
1	6	9	12	15	17	20	24	25	28	32	34	39	40	50	7
1	6	9	12	13	17	20	24	25	28	32	34	39	40	50	14

table 3.6.2 : result of ID3 algorithm for PCA data

these are sorted features

0.324	12
0.433	2
0.444	5
0.584	7
0.627	3
0.649	4
0.909	9
0.909	13
0.92	8
0.951	11
0.984	10
0.984	6
0.996	1

this is classification tree

2	4	7	12	13	16	19	22	25	28	31	35	39	70	1
1	4	7	12	13	17	19	23	25	30	32	35	39	70	15
1	5	8	11	15	18	21	23	25	29	33	35	38	50	7
2	6	9	11	15	17	21	23	27	28	33	35	37	50	3
1	6	9	12	15	18	21	22	26	29	33	35	39	50	13
1	5	7	11	13	17	19	23	25	28	31	36	39	50	2
3	6	8	10	15	16	21	24	27	30	33	36	37	50	4
1	6	8	10	15	18	20	22	27	29	32	36	38	50	6
1	6	8	10	14	16	19	23	25	30	32	36	38	50	9
1	6	9	12	15	18	21	22	26	30	32	36	39	50	14
3	6	8	11	14	16	20	23	27	30	33	34	37	70	5
1	4	7	12	13	18	19	22	25	29	32	34	38	70	8
1	6	8	12	14	18	20	23	26	30	33	34	39	70	10
1	4	7	12	13	18	19	22	25	29	32	34	39	70	11
1	4	7	12	13	18	20	22	25	29	31	34	39	70	12

table 3.6.3 : result of ID3 algorithm for RD data

[3]. If mean Si in hot metal is in range 5 and mean blow oxygen is in range 12 then corresponding class for converter life is in range 50.

[4]. If mean Si in hot metal is in range 5 and mean blow oxygen is in range 10 and mean basicity is in range 35 then the corresponding class for converter life is in range 50.

[5]. If mean Si in hot metal is in range 5 and mean blow oxygen is in range 11 and mean basicity is in the range 35 then corresponding class for converter life is in range 70.

[6]. If mean Si in hot metal is in range 5 and mean blow oxygen is in range 11 and mean basicity is in range 34 then the corresponding class for converter life is in range 50.

(iii) for RD data

The result of ID3 algorithm for RD data is presented in the table 3.6.3. The input to the algorithm is given in A7. Classification takes place in 3 steps and following decision rules are formulated.

[1]. If FeO in slag is in range 36 then the corresponding class for converter life is 50.

[2]. If FeO in slag is in range 34 then the corresponding class for converter life is 70.

[3]. If FeO in slag is in range 35 and mean Si in hot metal is in range 4 then the corresponding class for converter life is 70.

[4]. If FeO in slag is in range 35 and mean Si is in range 5 then corresponding class for converter life is 50.

[5]. If FeO in slag is in range 35 and mean Si is in range 6 then corresponding class for converter life is 50.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter a comparative study of all the methods is done. For comparison purpose, various modelling techniques are used which are developed by [7] and [13]. Actually, the data set finally obtained by using aforesaid techniques were given input to the models developed in [7] and [13].

First, a comparison is drawn between data prioritisation techniques. The variables selected from PCA, ICA, and experience of R&D persons of RDCIS,SAIL. The main idea behind using modelling as comparison ground is that if the selected variables are able to capture the variation in the data then the developed model will be able to predict the converter life with a good accuracy and can be used for future use. If we are unable to get a good model then it may be deduced that the total sample variation is not captured. Therefore, either we have to use a different set of variables or to increase the number of selected variables. Modelling is done using simple regression and auto regressive moving average in [13]and group method of data handling in [7]. Models are also developed using clusterwise regression in [13]in which all these clustering methods have been employed. Results are presented in terms of error statistics in the tables 4.1 to 4.10.

4.1 Comparison of Data Prioritisation techniques

From tables 4.1, 4.2 and 4.3 it is observed that best training in most of the cases takes place for ICA data. The mean training error in GMDH using ICA data is 1.5546 and mean prediction error is 6.5562. The training and prediction slope are 1.0046 and 1.0657 respectively. These quantities for PCA data are 4.7543, 6.198, 0.9981 and 0.9973. For Rd data mean training error is 5.9776 and prediction error is 5.1668. The training slope is 1.0052 and prediction slope is 0.9483. Similarly in regression also we are having best training and prediction statistics for ICA data followed by R&D data and then PCA data. In ARMA model also ICA data is giving best result. The PCA data is

data set used	training							prediction						
	mn error	ms error	rms error	error std	max error	min. error	slope	mn error	ms error	rms error	error std	max error	min. error	slope
rd	5.9776	0.5173	1.9949	4.0001	13.0343	0.2062	1.0052	5.1668	0.4005	4.475	3.6546	8.8214	1.5122	0.9483
ica	1.5546	0.0409	0.5839	1.2939	3.6989	0.0399	1.0046	6.5662	0.8133	6.3771	6.1822	12.7483	0.384	1.0657
pca	4.7543	0.3451	1.6292	3.45	10.2385	0.4216	0.9981	6.198	0.3849	4.3868	0.2705	6.4685	5.9275	0.9973

table 4.1 : comparision between ICA, RD and PCA for GMDH modelling

data set used	training							prediction						
	mn error	ms error	rms error	error std	max error	min. error	slope	mn error	ms error	rms error	error std	max error	min. error	slope
ica	2.02	0.09	0.82	2.17	6.56	0.04	1	2.4	0.06	17.7	0.73	3.13	1.67	0.98
rd	8.55	1.13	2.95	6.33	19.31	1.44	1.03	16.68	2.99	12.23	4.54	21.22	12.15	1.17
pca	2.38	0.07	0.72	1.06	3.76	0.36	0.98	47.46	23.51	34.28	9.9	57.36	37.66	1.47

table 4.2 : comparision between ICA, RD and PCA for regression modelling

data set used	training							prediction						
	mn error	ms error	rms error	error std	max error	min. error	slope	mn error	ms error	rms error	error std	max error	min. error	slope
ica	0.39	0	0.13	0.24	1.02	0.02	1	2.58	0.1	2.27	1.91	4.5	0.67	1.02
rd	8.24	1.06	2.98	6.21	20.06	0.3	1	19.93	4.51	15.02	7.35	27.28	12.58	0.93
pca	1.6	0.03	0.5	0.61	2.41	0.8	1.02	2	0.04	1.42	0.12	2.12	1.89	1

table 4.3 : comparision between ICA, RD and PCA for ARMA modelling

mn error = mean error, ms error = mean square error, rms error = root mean square error, error std = standard deviation of error, max error = maximum error, min error = minimum error

giving far better results than R&D data. Therefore, we can say that the variables selected from ICA are able to capture most of the variation present in samples. PCA and R&D experience are also able to capture the variation but not upto that extent to which ICA is able to capture.

4.2 Comparison between various clustering methods

All the clustering methods are compared using clusterwise regression. The results are presented in the tables 4.4 to 4.10. Clusterwise regression has been performed for 3 and 4 clusters using polynomial. All the three data sets are used in modelling. In most of the cases K-means clustering is giving best result for both 3 and 4 clusters. Fuzzy C-means clustering is also giving good result followed by Kohenon SOM and ART2 network. For Fuzzy ART network, though training statistics is comparable to other techniques but prediction is very bad. This means for Fuzzy ART network, modelled system is unable to capture the trend. This means the data partition is not good enough in this case. The other reason for bad results of Fuzzy ART network is that the sample used for prediction is always forming a separate cluster. So, the model is not able to fit a curve for this cluster and that is why we are having such bad results. The table for ID3 is provided separately because here we do not have any control over the number of cluster created. But we are having very good training here for ICA and PCA.

4.3 Conclusion

- Independent component analysis is best result for all modelling schemes; therefore we can say that independent components better capture the variation present in samples.
- Though for PCA data, training is always good but prediction is not up to the mark. This does not mean that the variables selected through PCA do not represent the variation but not to the extent to which ICA is capturing.
- In clustering techniques, K-means clustering and fuzzy C-means are giving better results, therefore we can say that data partition is better when we pre specify the number of clusters required.
- For Fuzzy ART prediction is worst because it is classifying the last sample in all the cases considered here (for symmetry in results) in a separate cluster.

clustering method used	training							prediction						
	mn error	ms error	rms error	error std	max error	min. error	slope	mn error	ms error	rms error	error std	max error	min. error	slope
k-means	0.106	0	0.037	0.078	0.287	0.018	1	4.62	0.32	3.997	3.258	7.877	1.362	1.033
Fuzzy c-means	0.085	0	0.029	0.064	0.249	0.003	1	5.696	0.398	4.459	2.707	8.403	4.875	1.072
kohenon	1.073	0.265	0.002	0.121	0.348	1.118	1	8.231	1.206	7.766	7.271	15.502	0.961	1.073
Art2	0.073	0	0.027	0.062	0.249	0.014	1	4.762	0.359	4.239	3.641	8.403	1.121	1.036
Fuzzy Art	0.266	0.001	0.089	0.178	0.533	0.014	1	50.166	50.001	50	49.834	100	0.331	0.498

table 4.4 : comparision of clustering methods used for ica data with 3 clusters

clustering method used	training							prediction						
	mn error	ms error	rms error	error std	max error	min. error	slope	mn error	ms error	rms error	error std	max error	min. error	slope
k-means	0.055	0	0.02	0.044	0.17	0.004	1	5.333	0.462	4.805	4.212	9.545	1.121	1.042
Fuzzy c-means	0.067	0	0.023	0.051	0.17	0.004	1	7.21	0.574	5.359	2.335	9.545	4.875	1.072
kohenon	0.181	0.001	0.075	0.202	0.599	0.015	1	4.213	0.314	3.96	3.69	7.903	0.523	1.043
Art2	0.085	0	0.029	0.063	0.249	0.014	1	6.639	0.472	4.857	1.764	8.403	4.875	1.066
Fuzzy Art	3.501	0.253	1.395	3.612	11.775	0.118	1.003	50.815	50.013	50.007	49.185	100	1.631	0.492

table 4.5 : comparision of clustering methods used for ica data with 4 clusters

mn error = mean error, ms error = mean square error, rms error = root mean square error, error std = standard deviation of error,max error = maximum error, min error = minimum error

clustering method used	training							prediction						
	mn error	ms error	rms error	error std	max error	min. error	slope	mn error	ms error	rms error	error std	max error	min. error	slope
k-means	0.07	0	0.025	0.058	0.214	0.005	1	8.174	1.14	7.551	6.872	15.046	1.302	1.082
Fuzzy c-means	0.07	0	0.025	0.058	0.214	0.005	1	8.174	1.14	7.551	6.872	15.046	1.302	1.082
kohenon	0.12	0	0.041	0.088	0.261	0.006	1	4.285	0.287	3.789	3.219	7.503	1.066	1.043
Art2	0.073	0	0.027	0.063	0.204	0.001	1	3.938	0.167	2.889	1.09	5.028	2.848	1.039
Fuzzy Art	1.81	0.046	0.593	1.139	4.897	0.544	1.001	100	100	70.71	0	100	100	0

table 4.6 : comparision of clustering methods used for pca data with 3 clusters

clustering method used	training							prediction						
	mn error	ms error	rms error	error std	max error	min. error	slope	mn error	ms error	rms error	error std	max error	min. error	slope
k-means	0.077	0	0.027	0.058	0.207	0.018	1	3.245	0.107	2.312	0.397	3.642	2.848	1.032
Fuzzy c-means	0.058	0	0.018	0.029	0.105	0.012	1	13.425	1.829	9.538	1.302	14.727	12.123	1.134
kohenon	0.102	0	0.042	0.087	0.262	0.002	1	5.015	0.488	4.939	4.862	9.876	0.153	1.049
Art2	0.073	0	0.027	0.063	0.204	0.001	1	30.938	0.107	2.889	1.09	5.028	2.848	1.039
Fuzzy Art	0.274	0.001	0.102	0.245	0.861	0.048	1	100	100	70.71	0	100	100	0

table 4.7 : comparision of clustering methods used for pca data with 4 clusters

mn error = mean error, ms error = mean square error, rms error = root mean square error, error std = standard deviation of error,max error = maximum error, min error = minimum error

clustering method used	training							prediction						
	mn error	ms error	rms error	error std	max error	min. error	slope	mn error	ms error	rms error	error std	max error	min. error	slope
k-means	0.073	0	0.024	0.047	0.176	0.011	1	8.338	0.712	5.967	1.307	9.644	7.031	1.034
Fuzzy c-means	0.0087	0	0.028	0.049	0.178	0.023	1	10.395	1.618	8.995	7.332	17.727	3.063	1.038
kohenon	0.418	0.004	0.168	0.44	1.359	0.024	1	2.699	0.091	2.13	1.338	4.036	1.361	0.973
Art2	0.071	0	0.024	0.049	0.176	0.01	1	8.642	0.757	6.152	1.002	9.644	7.64	1.086
Fuzzy Art	0.456	0.004	0.17	0.407	1.387	0.035	1	52.191	50.096	50.048	47.809	100	4.382	0.478

table 4.8 : comparision of clustering methods used for rd data with 3 clusters

clustering method used	training							prediction						
	mn error	ms error	rms error	error std	max error	min. error	slope	mn error	ms error	rms error	error std	max error	min. error	slope
k-means	0.056	0	0.019	0.04	0.126	0	1	6.239	0.505	5.026	3.405	9.644	2.833	1.034
Fuzzy c-means	0.077	0	0.025	0.046	0.178	0.016	1	6.883	0.62	5.566	3.82	10.709	3.063	1.038
kohenon	0.15	0	0.053	0.117	0.42	0.036	1	7.998	0.768	6.917	3.583	11.581	4.416	1.08
Art2	0.072	0	0.024	0.049	0.178	0.003	1	6.398	0.515	5.073	3.246	9.644	3.152	1.032
Fuzzy Art	0.456	0.004	0.17	0.407	1.387	0.035	1	52.191	50.096	50.048	47.809	100	4.382	0.478

table 4.9 : comparision of clustering methods used for rd data with 4 clusters

data set used	training							prediction						
	mn error	ms error	rms error	error std	max error	min. error	slope	mn error	ms error	rms error	error std	max error	min. error	slope
ica	0.013	0.002	0.0041	0.0074	0.0268	0.0027	1	7.59	0.63	5.61	2.321	9.911	5.269	1.0759
pca	0.0325	0.002	0.0148	0.0426	0.1419	0.0025	0.999	10.258	1.164	7.629	3.34	13.607	6.915	0.966
rd	0.062	0.005	0.0206	0 0409	0.125	0.001	0.999	32.13	18.47	30.395	28.5489	60.685	3.587	1.32

table 4.10 : error statistics of clusterwise regr. For ID3 clustering

mn error = mean error, ms error = mean square error, rms error = root mean square error, error std = standard deviation of error,max error = maximum error, min error = minimum error

4.4 Recommendation for Future work

A lot of data prioritisation and dimensionality reduction methods are available. Here, we have used PCA and ICA for feature prioritisation but these techniques can also be used for the transformation of the feature space. So, modelling should also be done using transformed feature space. May be we get better results. The experience can be validated using Saaty's method of feature prioritisation. This method has not been used here which should be used. Apart from this a lot of clustering techniques are also available, which can also be used.

References

[1] Dan W. Patterson " *Introduction to Artificial Intelligence and Expert System* " , Prentice - Hall of India Private Limited, 1992.

[2]Gail A. Carpenter & Stephen Grossberg " *Self-Organization of Stable Category Recognition Codes for Analog Input Patterns* ", Applied Optics Vol 26 No. 23 ,1 December 1987 , pp 4919-4930.

[3] Gail A. Carpenter & Sthphen Grossberg "*A Self Organizing Neural Network For Supervised Learning,Recognition and Prediction*", IEEE Communication Magazines September 1992, pp 38-53.

[4] I. T. Jollife " *Principal Component Analysis* " ,Springer Series in Statistics, Springer- Verlag .

[5] J.Karhunen,Errki Oja,Liuyue Wang,Ricardo Vigario and Jyrki Joutsensalo " *A Class of Neural Networks for Independent Component Analysis* " , IEEE Transactions on Neural Networks, Vol 8, No. 3 , May 1997, pp 486-503 .

[6] Kryzystof J. Cios & Ning Liu " *Machine Learning Method for Generation of a Neural Network Architecture: A Continuous ID3 Algorithm* " , IEEE Transactions on Neural Networks Vol 3 No.2 March 1992 pp 280-291.

[7] Neelamadhab Biswal "*Comparison of Various Layer Structured Modelling Techniques* ", M. Tech. Thesis, Deptt. Of EE,I.I.T. Kanpur,1998.

[8] Richard A. Johnson and Dean W. Wichern " *Applied Multivariate Statistical Analysis* " ,Prentice-Hall of India, 1996.

[9] S.Y.Kung "*Digital Neural Networks* ", PTR prentice Hall Inc. Newjersy, 1993.

[10] Simon Haykin " *Neural Networks : A Comprehensive Foundation*" Macmillan College Publishing Company, Newyork,1994.

[11] Sir Maurice Kendall "*Multivariate Analysis*", Second edition, Charles Griffin & Company Ltd, London, 1980.

.[12] T.J.Ross " *Fuzzy Logic with Engineering Applications* " , McGrawhill-Hill, Inc,Newyork,1995.

[13] V. Sharat Chandra "*Comparison of Different Modelling Techniques*", M.Tech. Thesis, Deptt. Of EE,I.I.T. Kanpur,1998.

Appendix

S.No.	Variable
1	HM wt.
2	HM Carbon
3	Si in HM
4	Mn in HM
5	Sulpher in HM
6	P in HM
7	Hm Temp
8	Scrap Wt
9	Blow O2
10	IST
11	Tap Temp
12	Tap-tap ti
13	Lime Addition
14	ore
15	dolo
16	lance ht
17	bath C
18	Bath Mn
19	Bath Si
20	Bath Sulpher
21	Bath P
22	Basicity in Slag
23	%FeO in Slag
24	%MgO in Slag
25	%CaO in Slag
26	%MnO in Slag

table A1: name of variables

campaign no	mean HM temp.	mean Blow O2	mean tap-tap time	mean Lime	mean ore	mean Bath C	mean S	mean P	mean Basicity	mean %FeO	mean %MgO	mean %CaO	mean %MnO	Actal no. of runs
1\10	1274.03	6513.96	102.251	9.63338	2.52558	0.090532	0.040618	0.046787	3.10974	20.1074	5.27128	48 3341	2.36154	772
1\11	1280.72	6467.77	99.7523	10.3934	3.00652	0.083028	0.039801	0.04761	2.95135	20.3338	5.12946	47.3065	1 82676	560
1\2	1268.12	6980.12	140.695	16.4524	1.42934	0.14872	0.052358	0.036517	2.63659	17.7688	3.74188	49.0289	3 01225	563
1\3	1252.12	7002.61	175.15	14.2833	1.15569	0.073785	0.042796	0 02564	2.47568	19.8751	3.40674	47.2282	2.94684	499
1\4	1232.97	6857.39	137.666	15.7592	1.5019	0.069777	0.048577	0.03707	2.49685	18.4437	3.05467	49.4806	1.84867	937
1\5	1228.11	7043.94	117.581	14.2333	2.12	0.089298	0.04296	0.033175	2.73586	20.4845	4.43195	48.6253	1.43083	595
1\6	1235.48	6964.83	134.779	13.0952	3.10135	0.079736	0.033374	0.04	2.50831	20.4511	5.31683	47.0311	1.07206	539
1\7	1247.9	6967	128.557	11.4379	2.25253	0.088591	0.037864	0.036909	2.6167	19.5651	4.79124	48.7944	1.35381	532
1\8	1259.56	6493.27	104.521	10.6612	1.82296	0.095519	0.036245	0.043714	2.67862	17.1683	4.84844	50.1312	2.16963	662
2\10	1278.53	6767.83	97.8564	10.7047	2.77052	0.076412	0.039485	0.045711	2.8035	21.4174	4 21158	47.21	1.50842	607
2\6	1242.52	6801.22	133.921	12.1535	2.71821	0.07925	0.034	0.037117	2.56382	18.6164	5.19177	49.2515	1.09374	712
2\7	1262.17	6419.67	116.784	10.4651	1.9778	0.097322	0.038628	0.038793	2.67557	18.6628	5.32563	48.8925	1.34292	724
2\8	1260.12	6319.54	113.064	10.027	1.94443	0.083917	0.03549	0.038336	3.18154	17.6008	5.4675	49.7808	2.705	746
3\4	1234.59	7011.5	140.178	12.6994	2.76217	0.073272	0.034192	0.033967	2.64884	20.7927	5.46568	47.5912	1.14136	546
3\6	1261.02	6519.13	106.516	10.532	1.97692	0.100815	0.038021	0.041075	2.65012	19.2193	5.2324	48.4078	1.55768	652

table A2: data sheet prepared for modelling after performing ica

campaign no	mean hm/(hm+scrap)	mean Si in hm	mean Mn in hm	mean blow O2	mean Tap. Temp	mean tap-tap tin	mean Lime	mean Ore	mean Bath C	mean S	mean P	mean Basicity	mean %FeO	mean %CaO	Actual no. of runs
1\10	0.922119	0.880105	0.694967	6513.96	1666.13	102.251	9.63338	2.52558	0.090532	0.040618	0.046787	3.10974	20.1074	48.3341	772
1\11	0.905884	1.0431	0.657711	6467.77	1670.65	99.7523	10.3934	3.00652	0.083028	0.039801	0.04761	2.95135	20.3338	47.3065	560
1\2	0.933459	1.18129	0.597939	6980.12	1671.9	140.695	16.4524	1.42934	0.14872	0.052358	0.036517	2.63659	17.7688	49.0289	563
1\3	0.942544	1.12535	0.580189	7002.61	1661.22	175.15	14.2833	1.15569	0.073785	0.042796	0.02564	2.47568	19.8751	47.2282	499
1\4	0.942501	1.16596	0.635481	6857.39	1661.03	137.666	15.7592	1.5019	0.069777	0.048577	0.03707	2.49685	18.4437	49.4806	937
1\5	0.908138	1.10912	0.546531	7043.94	1666.74	117.581	14.2333	2.12	0.089298	0.04296	0.033175	2.73586	20.4845	48.6253	595
1\6	0.904731	1.27833	0.707137	6964.83	1680.28	134.779	13.0952	3.10135	0 079736	0.033374	0.04	2.50831	20.4511	47.0311	539
1\7	0.910444	1.04208	0.627623	6967	1677.37	128.557	11.4379	2.25253	0.088591	0.037864	0.036909	2.6167	19.5651	48.7944	532
1\8	0.910769	0.924724	0.694303	6493.27	1679.42	104.521	10.6612	1.82296	0.095519	0.036245	0.043714	2.67862	17.1683	50.1312	662
2\10	0.90487	1.11022	0.550334	6767.83	1663.84	97.8564	10.7047	2.77052	0.076412	0.039485	0.045711	2.8035	21.4174	47.21	607
2\6	0.902376	1.14695	0.666119	6801.22	1673.24	133.921	12.1535	2.71821	0.07925	0.034	0.037117	2.56382	18.6164	49.2515	712
2\7	0.90675	0.932009	0.667626	6419.67	1673.87	116.784	10.4651	1.9778	0.097322	0.038628	0.038793	2.67557	18.6628	48.8925	724
2\8	0.904433	0.952568	0.717364	6319.54	1675.33	113.064	10.027	1.94443	0.083917	0.03549	0.038336	3.18154	17.6008	49.7808	746
3\4	0.910845	1.29996	0.699404	7011.5	1667.3	140.178	12.6994	2.76217	0.073272	0.034192	0.033967	2.64884	20.7927	47.5912	546
3\6	0.901154	0.935502	0.664253	6519.13	1668.77	106.516	10.532	1.97692	0.100815	0.038021	0.041075	2.65012	19.2193	48.4078	652

table A3: data sheet prepared for modelling after performing pca

campaign no	hm/(hm+scrap)	mean Si	%Si in hm >=1.1	mean Mn	mean blow oxygen	tap temp >=1700	tap-tap time >=70min	%yes slag coat.	mean lime add.	%C in ste <=0.05	%slag basicity <=2.5	%FeO >=22	mean MgO in slag	Actual no. of runs
1\10	0.922119	0.880105	10.8924	0.694967	6513.96	14.8489	63.2895	37.0466	9.63338	22.1264	2.5641	28.2051	5.27128	772
1\11	0.905884	1.0431	28.5714	0.657711	6467.77	19.0647	57.1429	44.6429	10.3934	27.49	13.5135	40.5405	5.12946	560
1\2	0.933459	1.18129	67.8363	0.597939	6980.12	24.7312	83.9416	42.984	16.4524	18.9024	40.9091	21.5385	3.74188	563
1\3	0.942544	1.12535	57.5	0.580189	7002.61	14.0496	89.5062	75.5511	14.2833	56.9892	46.3158	35.7895	3.40674	499
1\4	0.942501	1.16596	64.4444	0.635481	6857.39	17.6136	74.0458	45.4642	15.7592	55.1769	53.9326	14.4444	3.05467	937
1\5	0.908243	1.10899	48.0069	0.546706	7043.32	28.8793	72.4786	38.2403	14.2341	32.0683	28.5714	38.3459	4.43195	595
1\6	0.904731	1.27833	67.5522	0.707137	6964.83	30.137	87.8277	31.1688	13.0952	39.2276	48.5915	29.5775	5.31683	539
1\7	0.910444	1.04208	36.9048	0.627623	6967	28.1369	83 6852	46.8045	11.4379	39.7727	40.2062	24.7423	4.79124	532
1\8	0.910769	0.924724	17.7586	0.694303	6493.27	30.0687	66.0377	31.5254	10.6612	33.195	22.0183	5.50459	4.84844	662
2\10	0.90487	1.11022	48.6622	0.550334	6767.83	15.2174	60.6419	41.6804	10.7047	40.6186	20	47.3684	4.21158	607
2\6	0.902376	1.14695	55 7637	0.666119	6801 22	27.8351	78.4173	44.1011	12.1535	40.1667	49.2462	13.6364	5.19177	712
2\7	0.90675	0.932009	13.3903	0.667626	6419.67	26.383	68.8456	38.8122	10.4651	35.0413	24.0418	14.9826	5.32563	724
2\8	0.904433	0.952568	21.1749	0.717364	6319.54	26.4586	72 9252	31.7694	10.027	32 7893	7.69231	8.33333	5.4675	746
3\4	0.910845	1.29996	76.1639	0.699404	7011.5	27.234	86.7675	39.1941	12.6994	49.4888	33.5484	34.1935	5.46568	546
3\6	0.901154	0.935502	14.1066	0.664253	6519.13	21.6749	69.2427	42.0245	10.532	40.0347	28.5156	20.8661	5.2324	652

table A4 : data sheet prepared for modelling using experience of r&d persons

var1	1220-1245	1
	1245-1270	2
	>=1270	3
var2	6300-6600	4
	6600-6900	5
	>=6900	6
var3	90-125	7
	125-160	8
	>160	9
var4	9-12.0	10
	12.0-15.0	11
	>=15.0	12
var5	1.5-18	13
	1.8-2.5	14
	>=2.5	15
var6	.065-.08	16
	0.08-0.095	17
	>=0.095	18
var7	0.03-0.036	19
	0.036-0.04	20
	>=0.041	21
var8	0.025-0.03	22
	0.03--0.04	23
	>=0.04	24
var9	2.4-2.7	25
	2.7-3.0	26
	>=3.0	27
var10	17-18.5	28
	18.5-20.0	29
	>=20.0	30
var11	3.0-4.0	31
	4.0-5.0	32
	>=5.0	33
var12	47-48.5	34
	48.5-50.0	35
	>=50.0	36
var13	1.0-2.0	37
	2.0-2.8	38
	>=2.8	39
no. of runs	480-650	50
	>=650	70

variable number and corresponding ranges

output class	var1	var2	var3	var4	var5	var6	var7	var8	var9	var10	var11	var12	var13
70	3	4	7	10	15	17	20	24	27	30	33	34	38
50	3	4	7	10	15	17	20	24	26	30	33	34	37
50	2	6	8	12	13	18	21	23	25	28	31	35	39
50	2	6	9	11	13	16	21	22	25	29	31	34	39
70	1	5	8	12	13	16	21	23	25	28	31	35	37
50	1	6	7	11	14	17	21	23	26	30	32	35	37
50	1	6	8	11	15	16	19	23	25	30	33	34	37
50	2	6	8	10	14	17	20	23	25	29	32	35	37
70	2	4	7	10	14	18	20	24	25	28	32	36	38
50	3	5	7	10	15	16	20	24	26	30	32	34	37
70	1	5	8	11	15	16	19	23	25	29	33	35	37
70	2	4	7	10	14	18	20	23	25	29	33	35	37
70	2	4	7	10	14	17	19	23	27	28	33	35	38
50	1	6	8	11	15	16	19	23	25	30	33	34	37
70	2	4	7	10	14	18	20	24	25	29	33	34	37

table A5 : input to the ID3 algorithm for ica data

var1	0.9-0.92	1
	0.92-0.94	2
	>=0.94	3
var2	0.85-1.0	4
	1.0-1.2	5
	>=1.2	6
var3	.54-0.6	7
	0.6-0.67	8
	>=.67	9
var4	6300-6600	10
	6600-6900	11
	>=6900	12
var5	1660-1670	13
	1670-1680	14
	>=1680	15
var6	90-125	16
	125-160	17
	>=160	18
var7	9.0-12.0	19
	12.0-15.0	20
	>=15.0	21
var8	1.1-1.8	22
	1.8-2.5	23
	>=2.5	24
var9	0.065-0.08	25
	0.08-0.095	26
	>=0.095	27
var10	0.03-0.036	28
	0.036-0.041	29
	>=0.041	30
var11	0.025-0.03	31
	0.03-0.04	32
	>=0.04	33
var12	2.4-2.7	34
	2.7-3.0	35
	>=3.0	36
var13	17.0-18.5	37
	18.5-20.0	38
	>=20.0	39
var14	47.0-48.5	40
	48.5-50.0	41
	>=50.0	42
no. of runs	480-650	50
	>=650	70

variable number and corresponding ranges

output class	var1	var2	var3	var4	var5	var6	var7	var8	var9	var10	var11	var12	var13	var 14
70	2	4	9	10	13	16	19	24	26	29	33	36	39	40
50	1	5	8	10	14	16	19	24	26	29	33	35	39	40
50	2	5	7	12	14	17	21	22	27	30	32	34	37	41
50	3	5	7	12	13	18	20	22	25	30	31	34	38	40
70	3	5	8	11	13	17	21	22	25	30	32	34	37	41
50	1	5	7	12	13	16	20	23	26	30	32	35	39	41
50	1	6	9	12	15	17	20	24	25	28	32	34	39	40
50	1	5	8	12	14	17	19	23	26	29	32	34	38	41
70	1	4	9	10	14	16	19	23	27	29	33	34	37	42
50	1	5	7	11	13	16	19	24	25	29	33	35	39	40
70	1	5	8	11	14	17	20	24	25	28	32	34	38	41
70	1	4	8	10	14	16	19	23	27	29	32	34	38	41
70	1	4	9	10	14	16	19	23	26	28	32	36	37	41
50	1	6	9	12	13	17	20	24	25	28	32	34	39	40
70	1	4	8	10	13	16	19	23	27	29	33	34	38	40

table A6 : input to the ID3 algorithm for pca data

var1	0.9-0.92	1
	0.92-0.94	2
	>=0.94	3
var2	0.85-1.0	4
	1.0-1.1	5
	>=1.1	6
var3	10.0-35.0	7
	35.0-65.0	8
	>=65.0	9
var4	0.5-0.59	10
	0.59-0.66	11
	>=0.66	12
var5	6300-6600	13
	6600-6900	14
	>=6900	15
var6	13-19	16
	19-26	17
	>=26	18
var7	56-70	19
	70-80	20
	>=80	21
var8	30-40	22
	40-50	23
	>=50	24
var9	9.0-12.0	25
	12.0-14.0	26
	>=14.0	27
var10	17-30	28
	30-40	29
	>=40	30
var11	0-20	31
	20-40	32
	>=40	33
var12	6.0-20.0	34
	20.0-30.0	35
	>=30.0	36
var13	3.0-4.0	37
	4.0-5.0	38
	>=5.0	39
no. of runs	480-650	50
	>=650	70

variable number and corresponding ranges

output class	var1	var2	var3	var4	var5	var6	var7	var8	var9	var10	var11	var12	var13
70	2	4	7	12	13	16	19	22	25	28	31	35	39
50	1	5	7	11	13	17	19	23	25	28	31	36	39
50	2	6	9	11	15	17	21	23	27	28	33	35	37
50	3	6	8	10	15	16	21	24	27	30	33	36	37
70	3	6	8	11	14	16	20	23	27	30	33	34	37
50	1	6	8	10	15	18	20	22	27	29	32	36	38
50	1	5	8	11	15	18	21	23	25	29	33	35	38
70	1	4	7	12	13	18	19	22	25	29	32	34	38
50	1	6	8	10	14	16	19	23	25	30	32	36	38
70	1	6	8	12	14	18	20	23	26	30	33	34	39
70	1	4	7	12	13	18	19	22	25	29	32	34	39
70	1	4	7	12	13	18	20	22	25	29	31	34	39
50	1	6	9	12	15	18	21	22	26	29	33	35	39
50	1	6	9	12	15	18	21	22	26	30	32	36	39
70	1	4	7	12	13	17	19	23	25	30	32	35	39

table A7 : Input to the ID3 algorithm for rd data